VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Analýza vícevariačních sítí

Multivariate Network Analysis

2020 Tomáš Anlauf

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání diplomové práce

Student: **Bc. Tomáš Anlauf**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T025 Informatika a výpočetní technika

Téma: Analýza vícevariačních sítí
Multivariate Network Analysis

Jazyk vypracování: čeština

Zásady pro vypracování:

Cílem práce je implementace vybraných metod analýzy vícevariačních sítí a implementace uživatelského rozhraní pro interaktivní práci. Preferován je jazyk C#.

1. Rešerše obdobných řešení.
2. Implementace vybraných metod analýzy vícevariačních sítí.
3. Návrh a implementace (doporučeno webové) aplikace využívající implementované metody.
4. Dokumentace s využitím standardů softwarového inženýrství.

Seznam doporučené odborné literatury:

[1] Kerren, A., Purchase, H. C., Ward, M. O. (2014). Introduction to multivariate network visualization. In Multivariate Network Visualization (pp. 1-9). Springer, Cham.
[2] Van den Elzen, S., Van Wijk, J. J. (2014). Multivariate network exploration and presentation: From detail to overview via selections and aggregations. IEEE Transactions on Visualization and Computer Graphics, 20(12), 2310-2319.

Dále podle pokynů vedoucího práce.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **doc. Mgr. Miloš Kudělka, Ph.D.**

Datum zadání: 01.09.2019
Datum odevzdání: 30.04.2020

doc. Ing. Jan Platoš, Ph.D.
vedoucí katedry

prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 15. května 2020 ..... Anlauf .....

Souhlasím se zveřejněním této diplomové práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v magisterských programech VŠB-TU Ostrava.

V Ostravě 15. května 2020 ..... Anlauf .....

Rád bych poděkoval své rodině a přátelům, kteří mě podporovali a dále bych rád především poděkoval panu doc. Mgr. Miloši Kudělkovi Ph.D. za jeho ochotu a vedení při tvorbě této práce.

Abstrakt

Vícevariační sítě jsou tvořeny vrcholy a vztahy ve formě hran, ovšem dále obsahují data o těchto vrcholech a hranách ve formě atributů. Reálné sítě velmi často obsahují několik atributů a mnoho analytických úloh je závislých na analýze jak vztahů, tak i atributů. Většina metod pro analýzu se zaměřuje pouze na síťovou topologii zkonstruovanou z vícevariačních dat. Navíc jsou tyto metody omezené na určité domény a vyžadují odbornou znalost problematiky. Cílem této práce je navržení webového systému, které kombinuje známé metody pro průzkum a analýzu vícevariačních sítí, pro umožnění strukturální i vícevariační analýzy. Manuální či automatické tvorby skupin vrcholů a filtrace vrcholů na základě jejich hodnot atributů umožní uživateli získat náhled na síť a lépe jí tak porozumět. Nakonec budou prezentovány příklady analýzy datových sad, na kterých budou demonstrovány metody pro průzkum vícevariačních sítí.

Klíčová slova: vizualizace; vícevariační sítě; vykreslení grafu; interakce; vizuální analýza

Abstract

Multivariate networks are made up of nodes and relationships in a form of links, but also data about these nodes and links as attributes. Real-world networks are often associated with several attributes a many analysis tasks depend on analyzing both, relationships and attributes. Most analysis methods are focused only on network topology contructed from multivariate data. In addition, these methods are often domain specific and require expert knowledge. The aim of this thesis is to create an web explorative tool which combines well-known methods for multivariate network exploration and analysis to enable structural and multivariate analysis. The manual or automatic creation of selections and attribute-based filtration of nodes enable the user to gain insights and understand the network better. Finally, examples of dataset analysis are presented to demonstrate the methods for multivariate network exploration.

Keywords: visualization; multivariate network; graph drawing; interaction; visual analysis

Obsah

Seznam použitých zkratek a symbolů

SOM	–	Samoorganizační mapy
UML	–	Unified Modeling Language
MCVs	–	Koordinované vizualizační komponenty
MVC	–	Architektonický vzor Model-Pohled-Kontrolér
TCP/IP	–	Primární přenosový protokol/protokol síťové vrstvy
C#	–	Programovací jazyk C Sharp
HTML	–	Hypertext Markup Language
SVG	–	Škálovatelná vektorová grafika
CSS	–	Kaskádové styly
UCI	–	University of California, Irvine

Seznam obrázků

Seznam tabulek

1 Úvod

Průzkum, analýza a porozumění komplexním a rozsáhlým datovým sadám je náročnou výzvou. Mnoho z nich popisuje vztahy mezi objekty. Nalezení možností, jak tyto vztahy rozlišit v kontextu dalších souvisejících dat, je velmi důležité pro řadu různých vědeckých oblastí. Vizualizace je využívána pro náhled a porozumění mezi komplexními objekty a jejich vlastnostmi. Nejčastěji jsou tyto vztahy reprezentovány hranami, které spojují vrcholy, kde vrcholy zastupují určité objekty a hrany zobrazují, jaké mezi těmito objekty existují vztahy.

Nástroje pro vizualizaci se tedy potýkají s mnoha různými výzvami. Mezi tyto výzvy patří hlavně způsob, jak zvýšit srozumitelnost zobrazených dat pro uživatele a další problémy, týkající se rozsáhlých, komplexních a dynamických dat. V některých případech se vrcholy a hrany mohou překrývat i při vizualizaci menších datových sad, což může nastat v případě, že vrcholy grafu jsou hustě propojeny mezi sebou nebo je omezena velikost displeje. Mnohé algoritmy a nástroje se snaží docílit lepší čitelnosti těchto dat.

Většina algoritmů pro vykreslení grafu nebere v potaz, že vrcholy v síti mohou mít řadu dalších atributů, které mohou být důležité pro její vizualizaci. Výzkumníci z oblastí biologie, analýzy sociálních sítí nebo softwarového inženýrství čelí tomuto problému vizualizace sítí s dalšími daty. Jako příklad je možno uvést analýzu komunikace pomocí aplikací či emailu mezi uživateli z dané domény. Příkladem domény může být skupina přátel či celá organizace. Je zřejmé, že tito uživatelé mají určitý věk, pohlaví, charakteristiky vzhledu, adresu, plat, pracovní pozici atd. Všechny tyto atributy mohou být důležité pro vztahy v síti. Abychom mohli rozpoznat nějaké vzory ve vztazích, tak potřebujeme vizualizovat množinu atributů každého uživatele a podívat se, zda některé vzory v atributech neovlivňují vznik vztahů v této síti. Komunikují uživatelé ve stejné věkové kategorii a se stejným pohlavím více mezi sebou? Vytvářejí se více skupiny mezi uživateli s podobným platem či u souvisejících pracovních pozicí? Odhalení topologie sítě s těmito přídavnými daty může přispět k zodpovězení těchto otázek. Takovéto sítě, které obsahují doplňující atributy k jejich vrcholům či hranám, se obecně označují jako vícevariační.

Vizualizace vícevariačních dat je již tak výzva sama o sobě. Data se vyskytují v různých typech (kategoriální, numerické, ordinální atd.), proto jsou potřeba různé techniky a strategie pro jejich vizualizaci. Počet atributů představuje také významný problém. Jejich počet může být příliš velký pro jejich zobrazení vedle sítě.

Cílem této práce bude vytvoření webového systému, u kterého bude vynaložena snaha pro převzetí nebo vytvoření nových metod, které umožní co nejpodrobnější, ale zároveň přehledný průzkum sítě. Důraz bude kladen na co nejlépe čitelné rozložení sítě a na možnost průzkumu specifických částí se zobrazením detailů o hodnotách atributů vrcholů. Uživatel si v případě své potřeby bude schopen upravit rozložení a pozici vrcholů dle vlastních preferencí či označit oblasti zájmu, které může poté samostatně prozkoumávat.

Tato práce se rozdělena na dvě hlavní sekce. První sekce teoretické části se zabývá již existujícími podobnými řešeními, které lze v této oblasti nalézt. Druhá sekce pak rozebírá samotnou

definici vícevariačních sítí, popis jejich částí, reprezentaci sítě v počítači, algoritmy sloužící k převodu vektorových dat na síť, rozložení sítě při vizualizaci, náhledy na části sítě a úkony vykonávané nad těmito druhy sítě. Sekce třetí má za úkol vícevariační sítě propojit s jinými obory a tedy popsat, kde jsou využívány, případně kde by mohly být využity v budoucnu. Práce dále přechází do části praktické, která slouží jako úplná dokumentace, popisující vlastní implementaci nástroje pro vizualizaci a práci s vícevariačními sítěmi, včetně specifikace požadavků, popisu architektury, popisu použitých knihoven a technologií. Součástí praktické sekce této práce bude i část experimentální, kde budou popsány experimenty s navrženým systémem při práci s různými datovými sadami.

2 Související práce

Nejrozšířenějším způsob, jak vizualizovat vícevariační síť je pomocí grafu. Každý bod reprezentuje vrchol a pokud mezi nimi existuje hrana, tak je znázorněna pomocí úsečky nebo křivky. Práce jsou zejména zaměřené na způsob, jak vytvořit dvoudimenzionální rozložení pro graf, který zároveň co nejlépe vyjádří topologii sítě. Snaha je poté vynaložena na tzv. estetická kritéria, které nám umožní lepší čitelnost vizualizované sítě. Problém čitelnosti se zejména projevuje při vizualizaci sítí s velkým množství vrcholů nebo pokud daná vypočtená topologie obsahuje místa, kde jsou vrcholy hustě propojeny. Samotná velikost, barva, tvar, délka hran a další vizuální aspekty mohou být využity pro vyjádření hodnot atributů [1] [2] [3].

Hlavní inspirací pro tuto práci bylo řešení navržené Van den Elzen; Van Wijk [4]. Jejich metoda umožňuje průzkum a analýzu struktury sítě i vícevariačních dat, které jsou připojeny k vrcholům a hranám této sítě. Rozhraní systému je rozděleno na několik pohledů, sloužících pro zobrazení struktury sítě, manipulaci a zobrazení příslušných atributů a průzkumu vysokoúrovňového přehledu. Hlavní prvkem této metody je tvorba a úprava zájmových výběrů, které si uživatel zvolí dle svých preferencí a je schopen získat detaily o vrcholech v těchto výběrech a hranách mezi nimi. Tato práce byla inspirována metodou speciálně vytvořenou pro průzkum vícevaričních sítí, která je prováděná pomocí Sémantických jednotek [5].

Tyto Sémantické jednotky tvoří nepřekrývající se oblasti, každá reprezentující jeden kategoriální atribut. V každé oblasti jsou vrcholy rozmístěny dle své hodnoty atributu nebo jsou jejich pozice vypočteny dle rozmístění založeného na silách působících na vrcholy. Grafické rozhraní umožňuje zvolit si viditelnost hran, aby se zabránilo nepřehlednosti. Zvlášť může být zvoleno zobrazení hran v jednotlivých oblastech a mezi oblastmi. Výběry z první zmíněné metody jsou těmto Sémantickým jednotkám podobné, ovšem nejsou omezené pouze na jeden kategoriální atribut a umožňují lepší filtrování viditelných hran a zobrazení dat s nimi spojených.

Další metodou, která stále souvisí se zmíněnými výběry a je hojně využívaná k přehlednějšímu průzkumu, jsou čočky umožňující zaostření grafu [6] [7]. Čočky jsou využívány pro zobrazení hustých částí sítě a zobrazení více informací pro oblasti, které uživatele zajímají. Dále umožňují extrahovat část sítě pro hlubší průzkum [8].

Vedle metod, které se soustředí na topologické vlastnosti sítě, existují metody, které využívají vícevariační data k vypočtení rozložení, založené na atributech. Příkladem můžou být SOM (Samoorganizační mapy) [9], které umisťují vrcholy a hrany sítě na povrch koule nebo JauntNets [10], které umožňuje vykreslit atributy jako vrcholy kolem vrcholů sítě a vytvořit mezi nimi hrany, pokud hodnota atributu u vrcholu sítě přesáhne zadanou mez. Tyto způsoby umožňují lepší náhled na vícevariační data sítě. Navíc, vícevariační data mohou být použita pro definování rozložení s použitím grafu rozložení v rovině a následného vložení hran jako v systému GraphDice [11].

PivotGraphs [12] poskytuje agregovaný pohled na síť tím, že vrcholy umisťuje na dvoudimenzionální mřížku dle jejich hodnot kategoriálních atributů. Nevýhodou této metody je, že

nezachovává topologii sítě, což ztěžuje strukturální průzkum sítě. GraphTrail [13] také využívá agregovaný způsob vizualizace, který poskytuje zachycování interakcí uživatele a tuto historii integruje do pracovního prostoru pro průzkum. V této práci se využívají grafy či jiné struktury jako matice nebo tabulky pro umožnění průzkumu vícevariačních dat.

Kódováním na vrchol či hranu je možno dále zlepšit estetiku grafu. Je tím myšlena změna vzhledu např. barvy či velikosti vrcholu nebo hrany nebo zavedení jiných interpretací (grafy) vrcholů v grafu sítě [14]. Barvy jsou poté nejčastěji využity pro zakódování numerických či kategoriálních atributů. Existuje mnoho možností, jak lze reprezentovat vrchol mimo použití základních geometrických tvarů. Jako jeden z příkladů může sloužit použití grafů, ať už sloupcových, lineárních či jiných [15]. Dalším příkladem může být zobrazení menších sítí či komunit jako vrcholy [16]. A posledním příkladem z oblasti sociálních sítí je použití fotografií a značky zobrazené jako text [17].

Mnoho metod již bylo vytvořeno a cílem této práce bude tedy systém, u kterého bude vynaložena snaha pro převzetí nebo vytvoření nových metod, které umožní co nejpodrobnější, ale zároveň přehledný průzkum sítě. Důraz bude kladen na co nejlépe čitelné rozložení sítě a na možnost průzkumu specifických částí se zobrazením detailů o hodnotách atributů vrcholů.

2.1 Aplikace vícevariačních sítí

Vícevariční sítě jsou velmi rozšířenou formou dat v mnoha oborech. Nejčastěji jsou tyto sítě aplikovány v oborech analýzy sociálních sítí, biologických aplikacích a softwarovém inženýrství. Jejich využití ovšem není omezeno pouze na tyto obory. Jejich uplatnění se projevilo i v oborech jako oceánografie [18], systémové analýzy [19] a inženýrství [20]. V této sekci budou postupně popsány hlavně obory, u kterých je využití vícevariačních sítí běžné, jmenovitě tedy sociální sítě, biologie a softwarové inženýrství. Těmito obory se také zabývá práce od Nobre; Meyer; Streit; Lex [14].

2.1.1 Analýza sociálních sítí

Sociální sítě jsou sítě, kde vrcholy reprezentují sociální entity, např. lidi nebo organizace, a hrany určují jaký je mezi nimi vztah. Tyto sítě jsou velmi oblíbeným a dostupných zdrojem dat. Obor analýzy sociálních sítí se zabývá hledáním strukturálních vlastností sítě a analýzou příslušných atributů vrcholů a hran. Sociální sítě často také obsahují rozsáhlou sadu atributů, které jsou svázané s vrcholy či hranami. Právě toto velké množství atributů představuje jednu z nejhlavnějších výzev při vizualizaci takových sítí. Navíc se při takovéto analýze dopočítávají i další atributy, které jsou vypočteny z dané topologie sítě jako stupeň, centrality, shlukovací koeficienty, komunity atd.

Jelikož sociální sítě jsou obvykle označovány jako sítě malého světa [21], tak je pro jejich reprezentaci v systémech často využívaná matice sousednosti, pro její schopnost podpory hustých a silně propojených částí sítě. Příkladem takovýchto systémů může být MatLink [22] a Mat-

rixExplorer [23], které využívají matici sousednosti v hybridních nebo bok po bok umístěných pohledech. MatLink využívá maticové zobrazení s umožněním vykreslení hran po okrajích této matice a přidáním dynamického zvýraznění vztahů mezi vrcholy pro jednoduší průzkum lokálně hustých míst v síti. Tato modifikace je velmi užitečná pro úkony zahrnující hledání cest v topologii sítě.

Dalším příkladem podobným MatLinku je NodeTrix [24]. Tento systém využívá dva druhy zobrazení topologie sítě v jednou hybridním pohledu. Jejich řešení zobrazuje síť pomocí grafu, ale navíc tento graf upraví tak, že nahradí husté a silně propojené oblasti maticí sousednosti. NodeTrix také dovoluje uživatelům zakódovat atributy vrcholů a hran sítě pomocí následujících vizuálních prvků: barvy, průhlednost , tvar, výplň tvaru, barva hran, šířka a přidáváním značek.

Několik dalších přístupů ovšem využívá pouze grafy v tradičním smyslu pro vizualizaci topologie vícevariační sítě, a to i přesto, že vizualizace matice sousednosti je mnohem rozšířenější. GraphDice [11] zobrazuje mřížku jednotlivých dvojic atributů, kde každá pozice v této mřížce obsahuje graf s pozicí vrcholů, řízenou právě těmito atributy, a tak umožňuje uživateli zjistit, jak jsou hodnoty atributů distribuovány v síti. Vizster [17] zobrazuje pohled na graf topologie sítě, který je spojený s bočním panelem atributů a uživatel si může zobrazit detaily o vrcholech, o které má zájem.

Posledním příkladem je metoda, která efektivně kombinuje několik způsobů zakódování atributů v jednom pohledu [25]. Tento systém využívá paralelní pásma s vrcholy a hranami. Tímto přístupem jsou využívány atributy k rozdělení vrcholů do pásem, nebo-li kategorií, na základě jejich hodnoty atributu. V každé z těchto pásem může uživatel seřadit vrcholy podle zvoleného atributu. A nakonec lze na vrcholy zakódovat pomocí vizuálních prvků některý z vypočtených atributů, jako například stupeň vrcholu.

2.1.2 Biologické aplikace

Biologické sítě jsou další oblastí, ve které je často využívána vizualizace vícevaričních sítí. Sítě v této oblasti jsou nejčastěji charakterizovány vrcholy a hranami, které obsahují komplexní atributy z oblastí genetiky, výzkumu rakoviny a systémové biologie.

K vizualizaci vícevariačních sítí jsou zde především využívány grafy, na rozdíl od oblasti sociálních sítí, kde dominantní je vizualizace matice sousednosti. Častou oblastí zájmu u těchto sítí je porozumění, jak se atributy mění během biologicky důležitých průchodů. Tedy mnohé algoritmy se pokouší o vytvoření optimálního rozložení sítě, které je pro uživatele intuitivní [26] [27]. Takovéto optimální rozložení je poté velmi užitečné pro práci s vícevariačními sítěmi, která zahrnuje průzkum topologie a atributů biologických cest.

Existují i další techniky zaměřující se úkony spojené s průzkumem cest v biologické síti. Enroute [28] a Entourage [29] dovolují uživatelům zvýraznit cestu v síti a vytvořit detailní pohled na atributy dané cesty. Pathline [30] linearizuje množinu vstupních cest s přehledem atributů, umístěných vedle. Cerebral [31] používá několik koordinovaných pohledů pro zobrazení topologie a atributů sítě. Pohled na vrcholy a hrany rozděluje vrcholy podle jejich přesné pozice v buňce

a je připojen k několika dalším pohledům, které mohou ukazovat vývoj sítě a dále k pohledu na atributy.

Posledním příkladem je technika, která využívá spojených pohledů, zahrnujících matici sousednosti místo běžně používaného grafu [32]. V této práci je použita pootočená matice sousednosti, která obměňuje prvky na řádcích a sloupcích matice tak, aby vytvořila shluky nenulových prvků v blocích kolem diagonály.

2.1.3 Softwarové inženýrství

Vizualizace vícevariačních sítí v softwarovém inženýrství je součástí širší oblasti nazývané softwarová vizualizace. Softwarovou vizualizací je myšleno vizuální znázornění kterékoliv komponenty v životním cyklu softwaru. Toto zahrnuje zdrojové kódy, příslušnou dokumentaci, modely a vstupní a výstupní data. Přehled použití vícevariačních síti je uveden v [33] a [14].

V softwarové vizualizaci jsou sítě modelovány s vrcholy, které reprezentují některou z komponent softwaru, jako například třídy, soubory, knihovny, funkce a další. Hrany poté určují vztah mezi těmito komponenty, tedy mohou buďto reprezentovat hierarchii komponent, což může být hierarchie složek a souborů či funkcí ve třídách nebo vztah asociační mezi vrcholy, jako volání funkcí či tok dat. Atributy vrcholů a hran mohou být v této oblasti různé a často zahrnují vypočtené softwarové metriky jako například počet řádků kódu, počet zavolání funkce, počet tříd, čas běhu určitých modulů nebo funkcí.

Sítě jsou v oblasti softwarového inženýrství velmi rozsáhlé a obsahují velké množství atributů, proto je často kladen důraz na jejich škálovatelnost. Další charakteristikou sítí v tomto oboru je velký počet typů atributů, jelikož vrcholy a hrany mohou představovat různé entity softwarových komponent, lidí, dat, souborů, složek atd.

Unified Modeling Language (UML) diagramy jsou ve své podstatě grafy, kde atributy jsou zakódovány na vrcholy v podobě textu nebo značek. Jelikož sítě mají hierarchickou povahu, tak je vhodné použít kombinaci grafu a technik přetížení, tedy technik optimalizovaných pro zakódování skupin, do kterých komponenty patří nebo hierarchických vztahů mezi prvky sítě [34].

Stromové mapy jsou použity ke znázornění hierarchické povahy dat, od balíčků až po funkce. A jako obvykle, velikost, barva či výška a šířka objektů v prostoru můžou reprezentovat hodnoty atributů jednotlivých vrcholů.

Dále může být vícevariační síť použita k vizualizaci jednotlivých verzí softwaru. Zde je vizualizace využita ke sledování, jak se zdrojový kód měnil a vyvíjel během vývoje softwaru, což může vést k identifikaci a oddělení modulů, které jsou během vývoje konstantní a modulů, ve kterých jsou často prováděny změny.

2.1.4 Další oblasti aplikací

Dalšími oblastmi využívajícími techniky vizualizace vícevariační sítě jsou komunikační sítě, transportní sítě a bezpečnost.

Komunikační sítě se zabývají tokem informací mezi zařízeními a lidmi. Nejvíce využívanou formou vizualizace vícevariačních sítí v této oblasti jsou grafy [35]. Přesná poloha zařízení a lidí pak může sloužit k určení přesné pozice ve vizuální reprezentaci sítě. Dále mohou být vrcholy umístěny blíže k sobě pokud patří do stejné skupiny, jako například zařízení ve firmách.

Oblast bezpečnostních sítí se především zaměřuje na analýzu a vizualizaci, za účelem najít anomálie, které by mohly ukazovat na zranitelný prvek nebo pokus o útok. I tato oblast využívá často komunikačních sítí s tím rozdílem, že typy atributů a analytických úloh jsou spojeny spíše s neobvyklým chováním sítě nebo vzory v přenosu dat. Způsob vizualizace je různý a zahrnuje digramy vrchol-hrana s působícími silami [36], grafy jako paralelní osy [37] a matice [35].

U transportních sítí pak vrcholy představují lokace, jako města, země, křižovatky a hrany poté představují buďto přímé spojení mezi těmito vrcholy v podobě silničních nebo letových cest a nebo pohyb lidí a zboží po těchto cestách mezi vrcholy. Vrcholy mají typicky atributy, které popisují a reprezentují danou lokaci. Takovýmto atributem může být například počet obyvatel obce. Atributy hran jsou pak nejčastěji délka cesty, čas cesty nebo počet dopravních prostředků či lidí cestujících mezi lokacemi. Jelikož vrcholy představují reálné lokace, tak se jako podklad pro vizualizaci sítě často používají mapy, což znamená, že vrcholy budou zpravidla vždy mít pevné rozložení. Kvůli pevné pozici vrcholů se zde objevuje častý nepřehledný překryv hran. Jednou z metod řešící tento problém je spojení hran se společnými výchozími a cílovými vrcholy do jedné hrany, což zlepší čitelnost grafu a sníží velikost grafu [38]. Některé transportní sítě nevyužívají přesnou geografickou polohu, ale místo toho si zachovávají pouze přibližnou pozici a pořadí vrcholů, jako tomu je například u plánů metra [39].

2.2 Úlohy a dotazy vykonávané nad vícevariační sítí

Tato sekce se bude věnovat úlohám, které jsou spojeny s vícevariačními sítěmi. Za úlohu považujeme aktivitu, kterou by uživatel mohl chtít provést interakcí s vizuální reprezentací vícevariační sítě. Cílem uživatele je tedy získat náhled na data, která studuje.

Obecně pro průzkum dat byly specifikovány úlohy popsané v Amar; Eagan; Stasko [40]. Tyto úlohy jsou samozřejmě relevantní i pro grafy. Těchto úloh je celkově deset a jmenovitě jimi jsou:

- Získání hodnoty - Pro množinu objektů najdi jejich hodnoty atributů.
- Filtrace - Na základě nějaké podmínky najdi data, která tuto podmínku splňují.
- Výpočet derivované hodnoty - Pro množinu objektů vypočti agregovanou numerickou reprezentaci těchto objektů (např. průměr, medián atd.)

- Vyhledání extrémů - Najdi objekty, jejichž hodnoty daných atributů jsou extrémem přes všechny objekty
- Seřazení - Množinu objektů seřaď dle některé ordinální metriky.
- Určení rozsahu - Pro množinu objektů a daný atribut najdi rozsah hodnot tohoto atributu přes všechny objekty.
- Určení rozdělení atributu - Pro danou množinu objektů a daný numerický atribut charakterizuj rozdělení tohoto atributu.
- Nalezení anomálií - Identifikuj veškeré anomálie z dané množiny objektů se zohledněním určeného vztahu nebo očekávání.
- Shlukování - Pro množinu objektů najdi shluky podobných hodnot atributů.
- Korelace - Pro množinu objektů a dvou atributů urči užitečné vztahy mezi hodnotami těchto atributů.

Dále v práci od Valiati; Pimenta; Freitas [41] byly tyto úlohy rozřazeny do tří tříd: operační (jak je síť prezentována a prozkoumávána), analytické (jak jsou informace získávány ze sítě) a kognitivní (porozumění celé síti). Každá z těchto kategorií obsahuje jednu či více úloh. I přesto, že operační úlohy hrají důležitou roli ve vizualizaci relevantních informací, budeme se v této práci věnovat pouze úlohám analytickým. Tyto analytické úlohy zahrnují:

- Identifikace - Nalezení objektů a vlastností v datech. Úkony identifikace často zahrnují vyhledání objektů, které jsou v sítích sousedé s respektem ke struktuře sítě. Zahrnuje to také identifikaci podobností, rozdílů, vzorů, odlehlých měření, variací, vztahů a nesrovnalostí.
- Určení - Vypočtení derivovaných vlastností, které nejsou uvedeny v původních datech. Toto často zahrnuje výpočet statistických měr vlastností spojených s vrcholy nebo hranami. Zde jako příklad lze uvést výpočet sum, rozdílů, poměrů, průměrů, mediánů, rozptylů, směrodatných odchylek, koeficientů korelace a pravděpodobností. Dále zde zahrnujeme algoritmické výpočty derivovaných objektů, například shlukování.
- Přemístění - Znovu navštívení objektů nebo vlastností, které již byly identifikovány nebo vypočteny. Uživatel je si vědom existence těchto objektů a vlastností, ale musí vyvinout úsilí pro jejich znovu nalezení.
- Porovnání - Průzkum objektů či vlastností, které byly identifikovány nebo vypočteny, vůči sobě. Často to také zahrnuje úkon přemístění. Porovnání jsou využívána pro vyhledání podobností nebo rozdílů mezi vlastnostmi vrcholů či hran.

Úlohy pro analýzu vícevariačních sítí, které vycházejí z výše uvedených, byly popsány v Lee; Plaisant; Parr; Fekete; Henry [42] a ještě podrobněji popsány v [43]. Pro analýzu sítí jsou úlohy rozděleny do čtyř kategorií: *založené na topologii*, *založené na atributech*, *zaměřující se na průzkum* a *sloužící k získání přehledu*. Každá z těchto kategorií obsahuje několik úloh, které jsou velmi často žádané při analýze sítí.

Úlohy zaměřující se na průzkum obsahují pouze úlohy spojené s následováním cesty a opětovným navštívením již dříve navštívených vrcholů. Jelikož těchto úloh není mnoho a jsou často spojeny s ostatními třemi kategoriemi, tak není potřeba tyto úlohy příliš rozebírat, protože slouží pouze pro sběr informací o vrcholech a hranách.

2.2.1 Úlohy založené na topologii

Úlohy pro vyhledání sousedů kombinují analytické úlohy (identifikace, určení, přemístění a porovnání) k získání znalostí o sousednosti objektů. Pokud jsou vyhledány sousední objekty, tak uživatelé nejčastěji studují některou z jejich vlastností.

Úlohy pro dosažitelnost kombinují analytické úlohy (identifikace, určení, přemístění a porovnání) k získání znalostí o dosažitelnosti objektů. Objekt je dosažitelný z jiného objektu, jestliže mezi nimi existuje cesta libovolné délky, která je spojuje. Pokud jsou vyhledány dosažitelné objekty, tak uživatelé nejčastěji studují některou z jejich vlastností.

Úlohy pro vyhledání společných sousedů kombinují analytické úlohy (identifikace, určení a přemístění) k nalezení objektů, které sdílejí dva nebo více sousedů. Pokud jsou vyhledány sdílené objekty, tak uživatelé nejčastěji studují některou z jejich vlastností.

Úlohy souvislosti kombinují analytické úlohy (identifikace, určení a přemístění) k získání znalostí o souvislosti podsítí.

V Tabulce 1 jsou vypsány úlohy založené na topologii.

2.2.2 Úlohy založené na atributech

Úlohy pracující s vrcholy a hranami kombinují analytické úlohy (identifikace, určení a přemístění) k získání znalostí o vrcholech, hranách a jejich příslušných atributech. V Tabulce 2 jsou vypsány úlohy založené na atributech.

2.2.3 Úlohy sloužící k získání přehledu

Hledání charakteristik sítě bylo stanoveno jako jediná úloha pro získání přehledu [42]. Tato úloha zahrnuje určení velikosti sítě, rozdělení hodnot atributu v objektech nebo přibližné nalezení shluků v síti. Definice těchto úloh je založena na předpokladu, že vnější podpora (využití paměti, algoritmů) není dostupná během jejich průběhu a tak není možné získat přesné odpovědi a hodnoty, ale pouze jejich hrubý odhad [43]. Tyto úlohy naopak počítají s aktivním zapojením uživatele k získání informací. Jako příklad může sloužit způsob rozložení sítě při vizualizaci, kde

při určitém rozložení může být uživatel okamžitě schopen si udělat představu o tom, které části sítě budou tvořit shluky.

Úloha	Popis
Sousednost(objekty)	Nalezení množiny objektů sousedících s určeným objektem.
Sousednost(odvozená vlastnost)	Nalezení odvozené vlastnosti množiny objektů sousedících s daným objektem.
Sousednost(extrémní vlastnosti)	Nalezení entit s minimálním/maximálním počtem sousedů.
Dosažitelnost(objekty)	Nalezení množiny objektů dosažitelných z daného objektu.
Dosažitelnost(odvozená vlastnost)	Nalezení odvozené vlastnosti množiny objektů dosažitelných z daného objektu.
Dosažitelnost(objekty s omezením)	Nalezení množiny objektů dosažitelných z daného objektu s délkou cesty menší než n.
Dosažitelnost(vlastnosti s omezením)	Nalezení odvozené vlastnosti množiny objektů dosažitelných z daného objektu s délkou cesty menší než n.
Sdílení sousedé	Nalezení všech objektů, které jsou spojeny se všemi objekty z dané množiny objektů.
Souvislost(nejkratší cesta)	Určení, zda existuje cesta mezi dvěma objekty a nalezení nejkratší cesty mezi nimi.
Souvislost(kliky)	Nalezení klik v síti.
Souvislost(komponenty)	Nalezení počtu komponent sítě.
Souvislost(mosty, centrální prvky)	Nalezení objektů, které po jejich odstranění způsobí rozpad sítě na více komponent.

Tabulka 1: Úlohy založené na topologii

Úloha	Popis
Vrcholy(vlastnosti)	Nalezení vrcholů se specifickou hodnotou atributu.
Vrcholy(odvozená vlastnost)	Nalezení odvozené vlastnosti množiny vrcholů se specifickou hodnotou atributu.
Hrany(sousední vrcholy)	Nalezení vrcholů připojených k danému vrcholu hranou se specifickým atributem.
Hrany(extrémní hodnoty)	Nalezení vrcholu, který je incidentní s hranou s minimální/maximální hodnotou daného atributu.

Tabulka 2: Úlohy založené na atributech

3 Teoretický základ

Vícevariační síť je reprezentována grafem G (Sekce 3.1) a dále obsahuje n atributů (Sekce 3.2), které jsou připojeny k vrcholům či hranám [10]. Ukázka vícevariační sítě je zobrazena na Obrázku 1. Tato sekce tedy slouží k definici těchto sítí a k popisu částí, které jsou její součástí. Dále zde budou popsány základní struktury, které slouží pro reprezentaci sítě v počítači, následované metodami rozložení sítě při jejím vykreslení. Nakonec je zde popsána problematika operací a dotazů nad těmito sítěmi.

3.1 Charakteristika sítě

Graf (Jednoduchý graf) $G = (V, E)$ se skládá z konečné množiny vrcholů V a množiny hran E definované v 1 [44].

$$E \subseteq (u, v)|u, v \in V, u \neq v \tag{1}$$

Na základě těchto definicí je možné v literatuře najít řadu vlastností a charakteristik obecného grafu. Mezi ty nejdůležitější patří [44] [4] [45]:

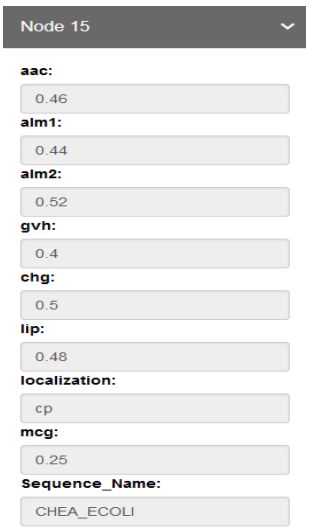
- Hrana $e = (u, v)$ a $u = v$ se nazývá smyčka.
- Pokud hrana e je obsažena v množině E vícekrát, tak je označována jako multihrana.
- Jednoduchý graf neobsahuje žádné smyčky a multihrany.
- Stupeň vrcholu v je počet vrcholů spojených hranou s vrcholem v.
- Orientovaný graf je graf s orientovanými hranami tj. (u, v) je uspořádanou dvojicí.
- Orientovaný graf je nazýván acyklický, pokud neobsahuje žádné orientované cykly tj. neexistuje žádná orientovaná cesta, ve které by byl stejný vrchol navštíven vícekrát.
- Graf je souvislý, pokud mezi každou dvojicí vrcholů u a v existuje cesta.
- Graf se nazývá rovinný, jestliže je možné ho vizualizovat v rovině s podmínkou, že se žádné dvě hrany grafu nekříží.

Dále v teorii sítí byla vyvinuta řada metrik, které mohou být použity k určení nejdůležitějších charakteristik topologie grafu jako například centrální aktéři v sociálních sítích. Tyto metriky mohou být aplikovány i na vícevariační sítě. Jedním z těchto přístupů je analýza komunit, která je založena na určitých shlukovacích metodách. Dalším příkladem mohou být centrality, které určují jak je daný vrchol nebo hrana v grafu důležitá. Centralita C je funkce, která přiřazuje hodnotu $C(u)$ vrcholu $u \in V$ daného grafu G. Na základě těchto přiřazených hodnot lze vrcholy porovnávat a určovat jejich důležitost.[44]. Nejjednodušším příkladem centrality je stupeň vrcholu v neorientovaném grafu.

3.2 Charakteristika atributů

Atributy $A = A_1, \ldots, A_n$ jsou připojeny k vrcholům (hranám). Atribut vrcholu A_i představuje jeden sloupec v tabulce atributů $A = (a_{ij})(j = 1 \ldots |V|; i = 1 \ldots n)$ [45]. Načež $a^v = (a_v 1, \ldots, a_v n)$ popisuje všechny hodnoty atributů pro vrchol v za předpokladu, že žádná data nechybí. V základu jsou atributy numerické, tedy vyjádřené pomocí číselných hodnot, či kategoriální, kde existuje definovaná konečná množina hodnot, které jsou přiřazovány jednotlivým záznamům. Pokud lze kategoriální data jasně uspořádat a lze snadno určit, která hodnota je větší a která menší, tak se jedná o data ordinální. Příkladem ordinálních dat mohou být dny v týdnu. Opakem jsou data nominální, kde jsou data neporovnatelná a nelze je tedy uspořádat. Jako příklad lze uvést krevní skupinu. Atributy mohou být také vypočteny z topologických vlastností sítě. Mezi tyto patří stupeň vrcholů a další centrality popsané v Sekci 3.1.

(a) Síť Ecoli s vyznačeným vrcholem číslo 15



(b) Atributy vrcholu číslo 15

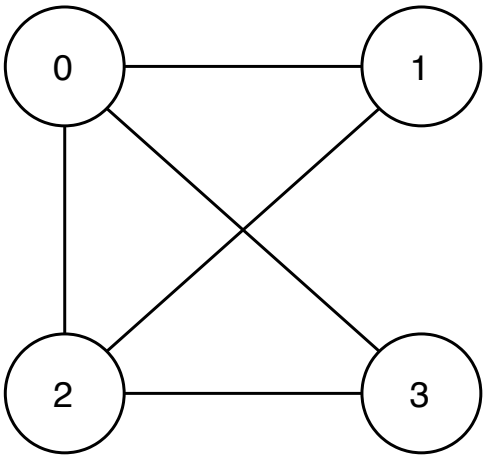
Obrázek 1: Ukázka vícevariační sítě

3.3 Reprezentace multivariační sítě v počítači

Abychom vůbec mohli s grafem G v počítači pracovat, tak je potřeba určit jak bude graf reprezentován v paměti. Existuje zde několik základních struktur, které lze k tomuto účelu využít. Jmenovitě se jedná o matici sousednosti, matici incidence, seznam hran a indexovaný seznam sousedů. Každá z uvedených struktur má své výhody a nevýhody a tedy neexistuje reprezentace, která by byla ve všech situacích nadřazená nad ostatními. Jednotlivé časové a paměťové složitosti operací nad uvedenými reprezentacemi neorientovaného grafu jsou znázorněny v Tabulce 3 [46].

	Matice sousednosti	Matice incidence	Seznam hran	Indexovaný seznam sousedů
Paměťová složitost	$O((\|V\|)^2)$	$O(\|V\| * \|E\|)$	$O(\|E\|)$	$O(\|V\| + 2 * \|E\|)$
Přidání nové hrany	$O(1)$	$O(1)$	$O(1)$	$O(1)$
Odebrání hrany	$O(1)$	$O(1)$	$O(E)$	$O(d_{max})$
Vyhledání hrany	$O(1)$	$O(1)$	$O(E)$	$O(d_{max})$
Vyhledání sousedů vrcholu v	$O(\|V\|)$	$O(\|V\| * \|E\|)$	$O(\|E\|)$	$O(\|1\|)$

Tabulka 3: Časové a paměťové složitosti operací nad jednotlivými reprezentacemi grafu, kde V je množina vrcholů grafu, E je množina hran grafu, d_max je maximální stupeň grafu



Obrázek 2: Příklad jednoduchého grafu pro znázornění reprezentace

3.3.1 Matice sousednosti

V matici sousednosti A každý řádek i sloupec odpovídá jednomu vrcholu. Existence hrany je poté zakódována na pozici určené daným řádkem a sloupcem. Neexistující hrana se často označuje hodnotou nula a naopak existující hodnotou vyšší než nula v závislosti na tom, zda je graf vážený či ne. Tato struktura funguje jak pro orientované sítě, kde řádky reprezentují výstupní vrcholy a sloupce zase ty vstupní, i pro neorientované sítě, u kterých je matice symetrická dle diagonály, a tudíž jsou hodnoty pod diagonálou redundantní. Matice sousednosti je obecně vhodná pro analýzu sousedství a shluků, ovšem není efektivní pro analýzu cest mezi vrcholy [14]. S pomocí matice sousednosti jsme schopni určit existenci, vložení a přidání hrany v konstantním čase. Hlavním problémem této struktury je, že uchovává hodnoty i pro neexistující hrany, což způsobuje velkou paměťovou složitost a tak je méně vhodná pro ukládání rozsáhlých a řídkých grafů. Příklad matice sousednosti, zobrazující graf na Obrázku 2, je uveden v Rovnici (2).

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \tag{2}$$

3.3.2 Matice incidence

I přesto, že reprezentace grafu pomocí matice incidence není použita v praktické části této práce, tak je zde uvedena pro úplnost, jako jedna z možností, jak k reprezentaci grafu přistupovat. Matice incidence I je velmi podobná matici sousednosti ovšem s tím rozdílem, že v matici incidence sloupce reprezentují jednotlivé hrany grafu namísto vrcholů. Hodnota na dané pozici této matice tedy určuje zda je vrchol s touto hranou incidentní či ne. Pro orientovaný graf jsou zde použity dvě různé hodnoty určující, zda je vrchol vstupní či výstupní. Tato reprezentace není tak efektivní pro základní operace nad grafy jako matice sousednosti [47]. Příklad matice incidence pro graf na Obrázku 2 je uveden v Rovnici 3.

$$I = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix} \tag{3}$$

3.3.3 Seznam hran

Pravděpodobně nejjednodušší způsob jak reprezentovat graf v počítači je pomocí seznamu hran. Jedná se tedy o pole, ve kterém je hrana reprezentována jako dvojice indexů vrcholů. Při práci s ohodnocenými grafy bude hrana reprezentována trojicí, kde třetí hodnota obvykle bývá vahou hrany. Jelikož jsou v poli obsaženy pouze hrany, tak je struktura mnohem méně náročná na paměť. Operace vyhledávání hran jsou ovšem pomalejší, protože při každém vyhledávání hrany musíme lineárně procházet celý seznam. Proto je seznam hran hlavně využíván pro perzistentní uchování sítě. Problém také nastává v případě uchovávání vrcholů, které nemají žádné sousedy. Takovýto vrchol nebude uveden v neupraveném seznamu hran nebo musí být uveden v jiné struktuře pro uchování existence vrcholů [48]. Příklad seznamu hran pro graf na Obrázku 2 je uveden v Rovnici (4).

$$L = (0,1),(0,2),(0,3),(1,2),(2,3) \tag{4}$$

3.3.4 Seznam sousedů

Poslední uvedenou strukturou je seznam sousedů. Každý z vrcholů si udržuje odkaz na seznam, který obsahuje veškeré jeho sousedy. Přístup k těmto seznamům probíhá v konstantním čase dle indexu příslušného vrcholu. Operace vyhledání existence hrany probíhá s lineární složitostí. Pokud jsou seznamy sousedů udržovány uspořádané, tak je možné zlepšit složitost vyhledávání na logaritmickou. V tomto tedy tato struktura není tak efektivní jako reprezentace pomocí matice, ale její výhodou oproti nim je výrazně nižší paměťová náročnost [48]. Příklad seznamu hran pro graf na Obrázku 2 je uveden v Tabulce 4.

Vrchol	Sousedé
0	$\{1, 2, 3\}$
1	$\{0, 2\}$
2	$\{0, 1, 3\}$
3	$\{0, 2\}$

Tabulka 4: Ukázka reprezentace pomocí seznamu sousedů

3.4 Algoritmy pro převod vektorových dat na síť

Množiny vektorových dat O nebývají vždy doprovázeny korespondující sítí, popisující vztahy mezi jednotlivými objekty. Proto pokud datová sada neobsahuje tyto informace, tak je potřeba síť vytvořit na základě vypočtených metrik mezi dvěma objekty. Metrika je tedy funkce ρ: $O \times O \rightarrow \mathbb{R}_+$. Nejčastěji využívané metriky jsou vzdálenosti mezi objekty a jejich podobnost. Výsledná síť by měla být nejlépe vytvořena tak, aby shluky, odlehlá měření, nejbližší sousedé a další vlastnosti byly zachovány. [49] Tato sekce tedy slouží pro seznámení s algoritmy, které řeší problém převodu vektorových dat na síť a zároveň jsou využity v praktické části této práce. Zvolenou metrikou pro všechny algoritmy používané v této práci je Gaussův kernel, jehož předpis je uveden v Rovnici (5).

$$K(o_i, o_j) = \exp\left\{-\frac{\|o_i - o_j\|^2}{2\sigma^2}\right\} \tag{5}$$

Objekty o_j, pro které je $\rho(o_i, o_j) > 0$ jsou *sousedé*. *Sousedství* objektu o_i je množina všech jeho sousedů. *Nejbližší soused* objektu o_i je objekt o_j, který je nejvíce podobný objektu o_i. Objekt o_i může mít více *nejbližších sousedů*. Hodnota σ je volitelná a obvykle se za ní dosazuje hodnota 1. [49]

3.4.1 Konstrukce ϵ a kNN grafu

ϵ-graf je neorientovaný graf, kde množina hran obsahuje dvojice objektů (o_i, o_j), pokud tedy hodnota metriky $\rho(o_i, o_j)$, pro ně vypočtené, nepřesahuje stanovený práh $\epsilon \in \mathbb{R}_+$. Mnoho efektivních algoritmů bylo vyvinuto pro určení optimální hodnoty prahu ϵ [50] [51], ovšem tento typ konstrukce snadno vede k tvorbě nesouvislých komponent a je tedy velmi náročné najít takovou hodnotu prahu, jehož výsledkem bude graf, který obsahuje vyhovující počet hran. [52]

Graf kNN nebo celým názvem *k-Neareast-Neighbours* je obecně orientovaný graf, kde hrana mezi objekty o_i do o_j vzniká v případě, že hodnota metriky $\rho(o_i, o_j)$ patří mezi k nejmenších hodnot množiny $\{\rho(o_i, o_k) | k = 1, \ldots, i-1, i+1, \ldots, n\}$. Tento typ konstrukce se ukázal být v praxi mnohem efektivnější než předchozí zmíněná konstrukce a proto již byla vedena řada výzkumů, které si dávaly za úkol tuto metodu zlepšit [53] [52]. Výhodou této metody je to, že hodnotou k jsme schopni nastavit minimální výstupní stupeň grafu a snadněji minimalizovat vznik nesouvislých komponent. [52]

Pro účely této práce bylo rozhodnuto, že výsledný algoritmus použitý v praktické části bude využívat kombinaci obou výše zmíněných metod. Časová složitost tohoto algoritmu je $O(n^2)$, kde $n = |O|$. Vstupními parametry je počet nejbližších sousedů vrcholů k a dále práh podobnosti ϵ. Pseudokód algoritmu je uveden v Algoritmu 1. Jednotlivé kroky algoritmu jsou:

1. Vypočti matici podobnosti S pro množinu objektů O.
2. Vytvoř množinu vrcholů V grafu G, kde vrchol v_i reprezentuje objekt o_i z množiny objektů O.
3. Vytvoř množinu hran E grafu G.
4. Množina E obsahuje hranu e_{ij} vrcholů v_i a v_j $(i \neq j)$, pokud vrchol v_j patří mezi jeho k nejbližších sousedů nebo je jejich podobnost vyšší než stanovený práh ϵ do množiny E.

Algoritmus 1: Algoritmus pro konstrukci kombinovaného ϵ-grafu a kNN grafu

Input: množina vektorových dat O, počet nejbližších sousedů k, práh podobnosti ϵ
Output: graf G

```
S ← CalculateSimilarityMatrix(O);
n ← len(O) ;
G ← CreateAdjancencyMatrix(n);
for i ← 0 to n do
    sortedObjectList ← SortObjectsBySimilarity(i, S);
    counter ← 0;
    for j ← 0 to len(sortedObjectList) do
        if counter < k and sortedObjectList[j] > ϵ then
            break;
        end
        G[i][sortedObjectList[j]] ← 1;
        G[sortedObjectList[j]][i] ← 1;
    end
end
```

3.4.2 *LRNet*

Následující algoritmus pro konstrukci ohodnoceného grafu byl publikován v Ochodkova; Zehnalova; Kudelka [49]. Je předpokládáno, že pro každý datový objekt lze vypočíst reprezentativnost, což je lokální vlastnost, která vychází z počtu objektů, které jsou nejbližšími sousedy zvoleného objektu. Hrany jsou vytvořeny mezi páry nejbližších sousedů a dále mezi individuálními objekty a to v takovém počtu, který odpovídá reprezentativnosti těchto objektů. Časová složitost tohoto algoritmu je $O(n^2)$, kde $n = |O|$.

Algoritmus LRNet je složen ze tří kroků:

1. Vypočti matici podobnosti S pro množinu objektů O.

2. Vytvoř množinu vrcholů V grafu G, kde vrchol v_i reprezentuje objekt o_i z množiny objektů O.

3. Vytvoř množinu hran E grafu G, kde E obsahuje hranu e_{ij} mezi vrcholy v_i a v_j $(i \neq j)$, pokud o_j je nejbližším sousedem o_i nebo o_j je reprezentativním sousedem o_i.

Pseudokód algoritmu LRNet je uveden v Algoritmu 2.

3.5 Rozložení sítě při vizualizaci

Rozlišujeme mezi třemi přístupy vizualizace sítě: Rozložení pomocí grafu, tabulkové rozložení a implicitní stromové rozložení [14]. Tabulkovým rozložením je především myšleno zobrazení pomocí matic či jejich variant. Implicitní stromové rozložení zakódovává vztah mezi vrcholy pouze jejich relativní pozicí, a tak jsou hrany reprezentovány pouze implicitně. V této sekci se zaměříme pouze na první ze zmíněných rozložení, jelikož je využito dále v praktické části.

3.5.1 Rozložení pomocí grafu

Grafy jsou nejběžnější grafické reprezentace sítí. V těchto grafech jsou vrcholy typicky vykreslovány jako jednoduché geometrické útvary (kružnice, čtverce atd.) a hrany jsou reprezentovány úsečkami či křivkami, které tyto vrcholy spojují. Tato rozložení jsou předmětem zájmu oboru kreslení grafu (graph drawing) a do dnešního dne bylo vyvinuto nespočet algoritmů řešících tuto problematiku.

Je rozlišováno mezi volnými rozloženími, kde pozice vrcholů není nijak omezena, příkladem může být rozmístění založené na silách působících na vrcholy a rozložením dle zvoleného stylu, kde pozice vrcholů je určena pomocí předdefinovaného schématu, jako například kruhové rozložení či rozložení v mřížce. Obě tyto rozložení lze poté zařadit do skupiny řízené topologií sítě. Jelikož je toto rozložení optimalizováno pro vizualizaci struktury sítě, tak samotná pozice vrcholů není schopna zakódovat jejich atributy. V tomto případě lze atributy vizualizovat tak, že pozměníme vzhled vrcholů či hran.

Další skupinou jsou rozložení řízené atributy, kde pozice vrcholů je určena pomocí atributu. U této metody lze volně měnit pouze způsob zobrazení hran, které spojují pevně umístěné vrcholy. Kategoriální data lze využít k rozdělení vrcholů do oblastí, každá odpovídající jiné kategorii. Samotné rozmístění vrcholů v těchto oblastech může být poté řešeno jiným způsobem. Numerické atributy nejčastěji určují přesné pozice vrcholů na ploše či v prostoru. Jako příklad může posloužit zeměpisná šířka a délka na mapě.

3.5.2 Kódování atributů na vrcholy a hrany

Kódováním atributů na vrcholy a hrany je myšlena modifikace vzhledu nebo zavádění značek na místo vrcholů a hran grafu [14], sloužící k vyjádření hodnot vybraných atributů.

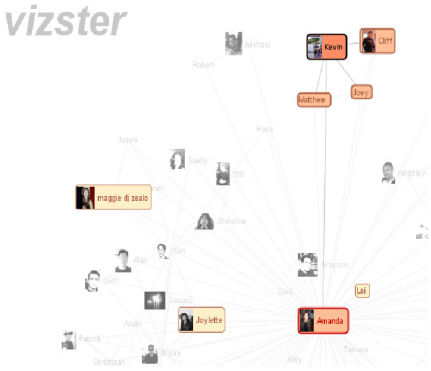
Algoritmus 2: Algoritmus LRNet

Input: množina vektorových dat O, počet nejbližších sousedů k, práh podobnosti ϵ
Output: graf G

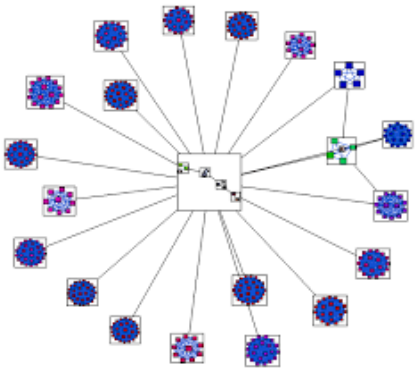
```
S ← CalculateSimilarityMatrix(O);
n ← len(O) G ← CreateAdjancencyMatrix(n);
degrees ← Dictionary ;
significances ← Dictionary ;
representativeness ← Dictionary ;
representativeNeighbours ← Dictionary ;
for i ← 0 to n do
    nearestNeighbour ← −1 ;
    maxSimilarity ← −1 ;
    for j ← 0 to n do
        if S[i][j] > 0 then
            degrees[i] ← degrees[i] + 1;
        end
        if S[i][j] > maxSimilarity then
            maxSimilarity ← S[i][j] ;
            nearestNeighbour ← j ;
        end
    end
    significances[nearestNeighbour] ← significances[nearestNeighbour] + 1;
end
for i ← 0 to n do
    if S[i][j] > 0 then
        representativeness[i] ← CalculateRepresentativeness(degrees[i], significances[i]);
    end
    else
        representativeness[i] ← 0 ;
    end
    k ← Round(representativeness[i] ∗ degrees[i]) ;
    potentialNeighbours ← SortObjectsBySimilarity(i, S);
    if k > 0 then
        for n ← 1 to k + 1 do ;                    // first element is the same as i

            G[i][potentionalNeighbours[i]] ← 1;
            G[potentionalNeighbours[i]][i] ← 1;
        end
    end
    else ;                                         // first element is the same as i

        G[i][potentionalNeighbours[1]] ← 1;
        G[potentionalNeighbours[1]][i] ← 1;
    end
end
```

U vrcholů se značky obvykle zobrazují jako text a jejich barva reprezentuje hodnotu některého z numerických nebo kategoriálních atributů. Dalším z častých kódování jsou různé tvary a ikony. Do tohoto je zahrnuto i používání grafů, např. krabicové grafy, histogramy atd. Vizster [17]

reprezentuje vrcholy pomocí malých fotek a značek (Obrázek 3a). Auber; Chiricota; Jourdan; Melançon [16] využívají toto kódování u sociálních sítí, zobrazující reprezentace podsítí pro určené dříve vypočtené topologické rysy (Obrázek 3b). Kódování může být realizováno pomocí komplexnější reprezentace vrcholů. Jako příklad může sloužit použití vizitek, na kterých jsou vypsány atributy pro určitou osobu, jako jméno, věk, foto atd.



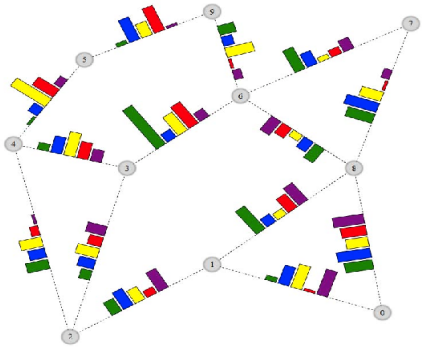
(a) Fotky a značky jsou zakódovány na vrcholy sociální sítě [17]



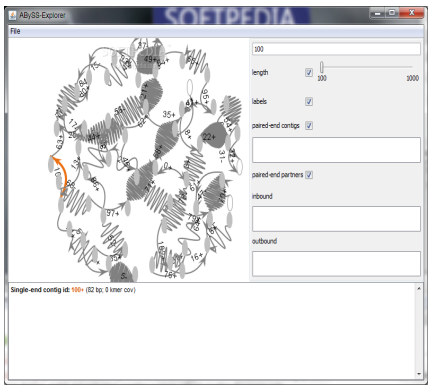
(b) Značky reprezentující podsítě sociální sítě [16]

Obrázek 3: Kódování atributů na vrcholy

U hran jsou atributy znázorněny pomocí modifikace vzhledu křivky, která hranu reprezentuje. Nejčastěji je u křivek upravována jejich šířka, barva, zakřivení, přerušování. Více atributů může být znázorněno pomocí sloupcových grafů [54] (Obrázek 4a) nebo vícebarevných segmentů s různou šířkou křivky [55]. Abyss Explorer [56] používá délku hrany k zakódování atributů hran (Obrázek 4b). Tento přístup ovšem může klást omezení na umístění vrcholů .



(a) Kódování atributů na hrany pomocí sloupcových grafů



(b) Kódování atributů na hrany pomocí délky hran

Obrázek 4: Kódování atributů na hrany

Kódování na vrcholy a hrany podporuje integraci úloh s topologií a atributy. Toto kódování je snadno srozumitelné pro uživatele a funguje dobře pro rozsáhlé komplexní sítě, vícevrstvé sítě i stromy. Rozsáhlost sítě ovšem často omezuje použití kódování. Při stoupajícím počtu vrcholů

musí být například velikost vrcholů limitována a tedy je obtížné v takovém případě použít tuto vizuální vlastnost k zakódování atributů. Kódování na vrcholy je doporučeno v případě, kde pouze malé množství atributů je zobrazeno na vrcholech a je možné zavést komplexnější kódování i na agregované vrcholy ke shrnutí vlastností těchto agregátů. Kódování na hrany je více omezené než na vrcholy. Hrany na rozdíl od vrcholů se často překrývají, což zhoršuje čitelnost kódování na hranách. Je doporučeno používat toto kódování pouze pro jeden numerický nebo kategoriální atribut.

3.6 Vizualizační komponenty

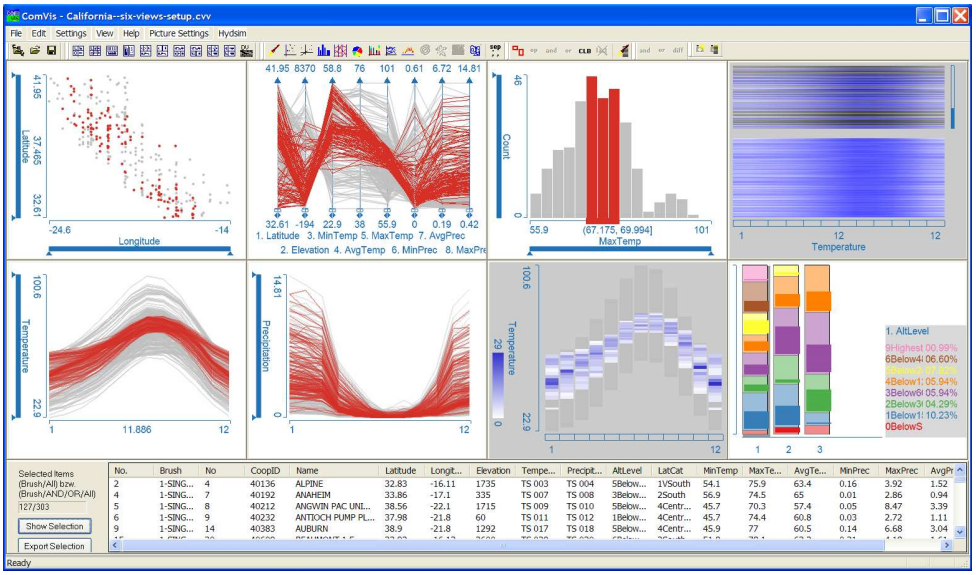
Při vizualizaci vícevariační sítě je obvykle využita technika koordinovaných vizualizačních komponent (MVCs). Tento přístup používá oddělené komponenty pro práci s atributy a topologií sítě. Typickým příkladem je kombinace grafu s technikami multidimenzionální vizualizace dat [57] nebo zobrazení komponent pro zobrazení detailů o jednotlivých vrcholech [17]. Rozlišujeme mezi třemi druhy koordinovaných vizualizačních komponent: umístěné vedle sebe, integrované a přetížené [14] [58].

3.6.1 Umístění vizualizačních komponent vedle sebe

Při využití této techniky se vizualizace topologie sítě a vizualizace atributů oddělí do dvou nebo více komponent. Toto činí tento způsob umístění flexibilním a jednoduchým na implementaci. Vztahy mezi informacemi v jednotlivých komponentech nejsou nijak vizuálně vyjádřeny a často jsou odhaleny až při interakci. Hlavní výhodou při použití tohoto umístění je schopnost každé vizualizační komponenty se zabývat pouze určitou částí sítě a soustředit se na vykonávání úloh spojených s touto částí co nejlépe. Tento způsob umístění je široce využíván při vizualizaci vícevariačních sítí [17] [4] [11]. Dobře navržené umístění komponent vedle sebe umožňuje uživateli se lépe v systému orientovat při analýze dat. Nalezení ideálního rozmístění vizualizačních komponent může ovšem představovat náročnou výzvu. [59].

Jako příklad zde poslouží systém ComVis [60], což je multidimenzionální vizualizační systém, který využívá několika vizualizačních komponent umístěných vedle sebe pro průzkum komplexních datových sad. Na Obrazku 5 je zobrazena ukázka průzkumu meteorologických dat pomocí systému ComVis. Ve spodní části systému je zobrazena datová sada ve formě tabulky. Uživatel v tomto případě vytvořil osm komponent. Každá z těchto komponent je využita k vykreslení jiného grafu. Dále uživatel označil několik sloupců v histogramu. Tato vyznačená data jsou poté systémem zvýrazněna ve všech ostatních grafech. Důležité je si povšimnout, že zde neexistuje žádné viditelné spojení korespondujících prvků mezi jednotlivými komponentami, což je typickým rysem vizualizačních komponent umístěných vedle sebe, který je odlišuje od komponent integrovaných.

Umístění vizualizačních komponent vedle sebe je doporučeno v případě, že se pracuje s rozsáhlými sítěmi a velkým množstvím nebo různorodými typy atributů vrcholů a hran. Každou



Obrázek 5: ComVis [60] (Vizualizační komponenty umístěné vedle sebe). Vizualizace meteorologické datové sady

komponentu lze optimalizovat pro práci buďto s topologií nebo s atributy, což umožňuje podporu nezávislé analýzy obou částí vícevariační sítě. Pokud je potřeba, tak je možné vytvořit spojení mezi těmito komponenty pomocí selekce vrcholů a hran, ale i přesto je poté složité identifikovat korespondující data mezi jednotlivými náhledy. Nejsou proto vhodné pro úlohy týkající se topologické struktury.

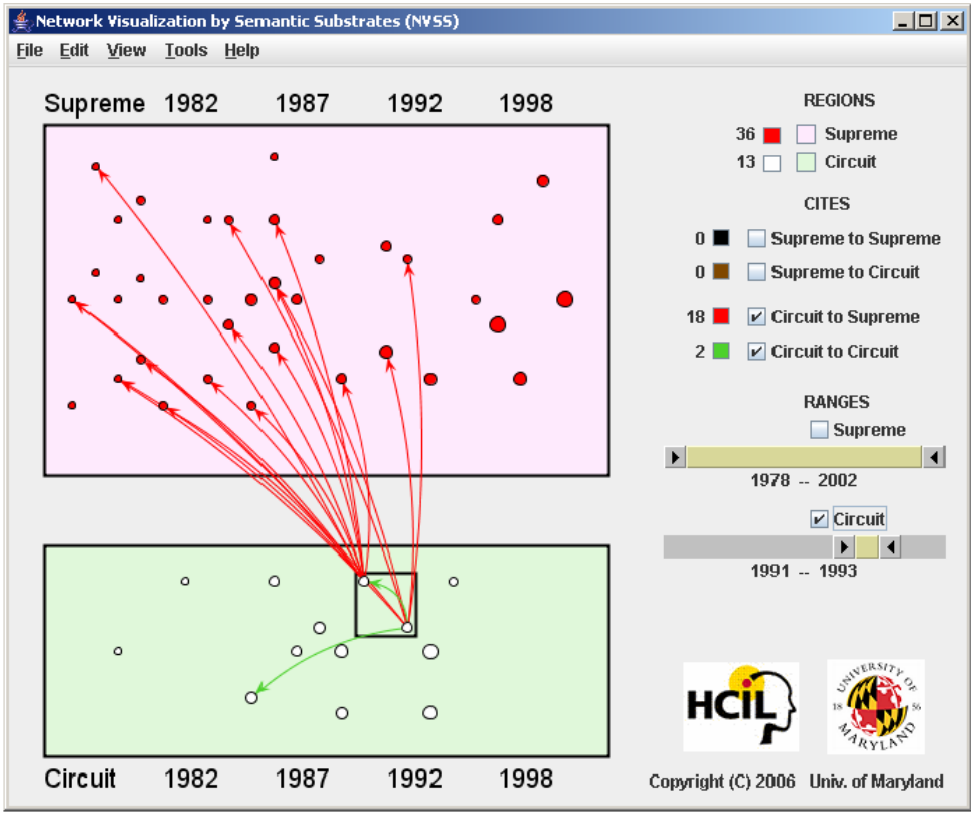
3.6.2 Integrované vizualizační komponenty

Integrované vizualizační komponenty vycházejí z komponent umístěných vedle sebe. Vizuální kompozice je v podstatě identická s tímto typem umístění. Hlavním rozdílem je, že integrované vizualizační komponenty jsou umísťovány s ohledem na ostatní komponenty a jejich propojení není implicitní, nýbrž jsou vztahy mezi prvky jednotlivých náhledů explicitně vykresleny, například pomocí úseček či křivek.

Využití explicitního vyjádření vztahů mezi integrovanými vizualizačními komponenty napomáhá porozumění vztahů mezi prvky komponent, ale při větším množství těchto vztahů může docházet k hromadění grafické reprezentace vztahů, což způsobí vážný problém nečitelnosti těchto vztahů. Nejběžnější strategií jak se tomuto vyhnout je pomocí agregace vztahů nebo zobrazení pouze zvolených vztahů.

Příkladem použití integrovaných vizualizačních komponent mohou být Sémantické jednotky [5], které jsou ukázány na Obrázku 6. Sémantické jednotky jsou nepřekrývající se oblasti, ve kterých jsou vykresleny vrcholy na základě některého vybraného kategoriálního atributu. Pozice každého vrcholu je poté určena dle konkrétní hodnoty atributu. Tato technika se zařazuje do

integrovaných vizualizačních komponent, jelikož je možné přidat vizuální propojení mezi jednotlivými oblastmi.



Obrázek 6: Sémantické jednotky [5] (Integrované vizualizační komponenty). Vizualizace sítě datové sady soudních případů pomocí Sémantických jednotek

Integrované vizualizační komponenty jsou velmi dobré v integrování komplexních atributů s topologií sítě, pokud topologie je reprezentována rozumně v lineárním rozložení. Tyto komponenty nejsou obvykle schopny vizualizovat komplexní topologii sítě, ale mohou být velmi užitečné při použití linearizace, např. uživatelem definovanými cestami. Na rozdíl od komponent umístěných vedle sebe, integrované komponenty excelují při práci s cestami a sousedstvími.

3.6.3 Přetížené vizualizační komponenty

Tento vzor je charakterizován kompozicemi, ve kterých je jedna vizualizace nazývaná klient, vykreslena uvnitř jiné vizualizace nazývané hostitel a využívá stejného prostorového mapování jako hostitel. Výsledná vizualizace se potom stane kombinací vizualizací jednotlivých komponent. Často se u vybraných komponent nastavuje průhlednost, která umožní uživateli snadno vidět všechny překryté komponenty. Klasickým příkladem je zobrazení na mapě, kde města patří do stejného shluku (státu) a hranice mezi státy představují hrany mezi nimi.

GeoSpace je systémem využívajícím přetížené vizualizační komponenty k průzkumu komplexních datových vrstev. Systém dovoluje překrytí několika datových sad dle požadavků uživatele.

Na Obrázku 7 je ukázka systému GeoSpace s vizualizací míst ve městě Cambridge, ve kterých došlo ke zločinu. Místa zločinu jsou reprezentovány červenými body, které jsou umístěny na mapě města Cambridge v místech, kde byly zločiny nahlášeny.

Obrázek 7: GeoSpace [61] (Přetížené komponenty). Zobrazení zločinů na geografické mapě města Cambridge

Přetížené vizualizační komponenty jsou vhodné pro zobrazení množin, skupin či shluků na již existující reprezentaci topologie sítě. Tato metoda nejlépe funguje, když prvky ze stejných shluků se nachází blízko sebe, což je běžný případ při práci se shluky. Hlavním omezením tohoto přístupu je omezený počet atributů, které je možno vizualizovat. Zakódování jednoho či dvou atributů najednou je možné, ale při větším počtu začne přetížení způsobovat nečitelnost vizualizace. Je proto doporučeno používat přetěžovaní v případě potřeby vizualizace shluků s použitím grafu jako podkladu.

3.7 Hledání komunit

Komunity, také nazývané jako shluky nebo moduly, jsou skupiny vrcholů, které mají podobné vlastnosti nebo hrají v grafu podobnou roli [62]. Grafy vycházející z reálných dat často vykazují známky komunitní struktury. Společnost, ve které žijeme, tuto strukturu následuje s širokým množstvím možností, jak skupiny organizovat: rodiny, přátelské vztahy, vesnice, města, národy. Na Internetu si lze také nalézt virtuální skupiny, nazývané jako online komunity. Výskyt komunit se také může objevit v síťových systémech z oborů biologie, informatiky, ekonomiky, politiky atd.

Detekce komunit je velmi důležitá pro identifikování modulů a jejich hranic. Toto umožňuje klasifikaci vrcholů dle jejich strukturální pozice v modulu. Tedy vrcholy, které se nacházejí v centrech těchto modulů a sdílejí velké množství hran s vrcholy ze stejné komunity, lze považovat za důležité pro řízení a stabilitu uvnitř komunity. Naopak vrcholy ležící na hranicích mezi moduly mají důležitou roli zprostředkovatelů vztahů a komunikace s jinými komunitami [62]. Komunity nám navíc umožní zlepšit rozložení grafu při jeho vizualizaci a to tak, že jednotlivé moduly je možno od sebe barevně a vzdáleností oddělit a tím docílit jeho lepší čitelnosti.

Neexistuje jednotná definice pro komunitu. Její definice je často závislá na specifickém systému a aplikaci. Dále bude uvedeno několik základních definic pro komunitu [62]. Často se stává, že komunity jsou algoritmicky definovány nebo-li jsou pouze konečným produktem algoritmu.

Komunitu lze tedy matematicky považovat za podgraf C grafu G, kde $N_c = |C|$ a $N = |G|$. Dále definujeme vnitřní a vnější stupeň vrcholu $v \in C$, vnitřní stupeň jako počet hran spojujících vrchol v s ostatními vrcholy z komunity C a vnější stupeň jako počet hran spojujících vrchol v s vrcholy ze zbytku grafu. Pokud se vnější stupeň vrcholu rovná nule, tak vrchol v má za sousedy pouze vrcholy z komunity C, což značí, že vrchol v je umístěn ve vhodné komunitě. Naopak pokud je vnitřní stupeň roven nule, tak je vrchol v odpojen od komunity C a měl by být přiřazen do jiné komunity. Vnitřní stupeň komunity C je součet všech vnitřních stupňů jejích vrcholů. Opačně potom, vnější stupeň komunity C je součet všech vnějších stupňů jejích vrcholů. Celkový stupeň je poté získán jako součet všech stupňů vrcholů komunity C.

V Rovnici (6) definujeme vnitřní hustotu shluku $\delta_{int}(C)$ komunity C jako poměr mezi počtem vnitřních hran v C a počtem všech možných vnitřních hran.

$$\delta_{int}(C) = \frac{\text{počet vnitřních hran v C}}{N_c(N_c - 1)/2} \tag{6}$$

Dále mezishluková hustota $\delta_{ext}(C)$ je definována v Rovnici (7) jako poměr počtu hran vedoucích z vrcholů z komunity C do zbytku grafu a maximálním možným počtem mezishlukových hran.

$$\delta_{ext}(C) = \frac{\text{počet mezishlukových hran z C}}{N_c(N - N_c)} \tag{7}$$

Aby C byla komunita, tak $\delta_{int}(C)$ by měla být značně vyšší než hustota δ_G celého grafu G, která je definována jako poměr počtu hran v grafu G ku maximálnímu možnému počtu hran $N(N-1)/2$. Na druhou stranu, $\delta_{ext}(C)$ musí být menší než δ_G. Cílem většiny shlukovacích algoritmů je tedy najít nejlepší možnou kombinaci co nejvyššího $\delta_{int}(C)$ a co nejmenšího $\delta_{ext}(C)$.

Vyžadovanou vlastností komunity je souvislost. Aby C byla komunitou, tak očekáváme, že bude existovat cesta mezi každými dvěma vrcholy vedoucí pouze před vrcholy z C. Hledání komunit je proto na grafech s více nespojitými komponenty zjednodušeno, protože stačí analyzovat každou komponentu zvlášť.

S těmito definovanými základními požadavky můžeme uvést definice komunity [62]. Existuje mnoho definic z oblasti sociálních sítí, informatiky či fyziky. Rozlišujeme mezi třemi typy definic: lokální, globální a založené na podobnosti vrcholů.

Lokální definice nám říkají, že komunity jsou součásti grafu s velmi malým počtem vazeb na zbytek systému. Zaměřují se na podgraf a jeho blízké sousedství a zbytek grafu zanedbávají. V analýze sociálních sítí byly zavedeny čtyři kritéria: celková propojenost, souvislost, stupeň vrcholu a porovnání vnitřní a vnější propojenosti. Korespondující komunity jsou nejčastěji maximální podgrafy, které nemohou být rozšířeny o další vrcholy bez ztráty vlastnosti, která je definuje. V případě celkové propojenosti mluvíme o klikách, nebo-li podgrafech, ve kterých existuje hrana mezi každými dvěma vrcholy. U souvislosti hledáme n-kliky, podgrafy, kde vzdálenost vrcholů není větší než n. Myšlenkou pro stupně vrcholů je hledat takový podgraf, kde všechny vrcholy mají stupeň roven či vyšší než k. A nakonec můžeme hledat komunity označované jako silné, ve kterých mají všechny vrcholy v rámci komunity vyšší stupeň vnitřní oproti tomu vnějšímu.

3.7.1 Modularita

Mnoho algoritmů je schopno objevit optimální podmnožinu rozdělení do komunit. Zpravidla je lepší, aby tato podmnožina byla co nejmenší, aby se uživateli usnadnil výběr mezi vhodnými rozděleními. Jiné techniky, jako ty založené na hierarchickém shlukování, jsou schopné nalézt velké množství dělení grafu. Neznamená to ovšem, že tyto shluky nejsou stejně dobré. V takovýchto případech je tedy vhodné mít kvantitativní kritérium, které určuje, jak je rozdělení grafu kvalitní. Funkce kvality [62] je funkce, která přiřazuje číslo každému nalezenému shluku. Shluky s vysokou hodnotou jsou považovány za dobré. Je důležité si uvědomit, že nelze vždy jednoznačně určit, zda jedno dělení grafu je lepší než druhé. Toto vždy záleží na tom, jak je komunita pro daný problém definována a na tom, jakou funkci kvality využíváme.

Nejpoužívanější funkcí kvality je modularita. Je založena na myšlence, že u náhodného grafu neočekáváme, že bude mít komunitní strukturu. Srovnáním hustoty podgrafu komunity s hustotou stejné skupiny vrcholů, ale s náhodně přepojenými hranami, můžeme určit, zda graf komunity lze považovat za hustý nebo je jeho propojení náhodné.

Mějme tedy síť s N vrcholy a L hranami a s rozdělením do n_c komunit a každá komunita má N_c vrcholů, které jsou spojeny L_c hranami a $c = 1, \ldots, n_c$. Pokud L_c je vyšší než očekávaný počet hran mezi N_c vrcholy s danou posloupností stupňů sítě, tak vrcholy podgrafu C_c mohou být součástí skutečné komunity [63]. Rozdíl mezi skutečným propojením vrcholů sítě (A_{ij}) a očekávaným počtem hran mezi i a j, pokud síť je náhodně propojená (p_{ij}), je určen v Rovnici (8).

$$Q_c = \frac{1}{2L} \sum_{(i,j) \in C_c} (A_{ij} - p_{ij}) \tag{8}$$

Zde může být p_{ij} určeno náhodným přepojením původní sítě se zachováním stupně každého vrcholu. Použití null modelu, zachovávajícího stupně vrcholů, je uvedeno v Rovnici (9).

$$p_{ij} = \frac{k_i k_j}{2L} \tag{9}$$

Pokud je Q_c kladné, pak podgraf C_c má více hran než bylo očekáváno, tudíž můžeme ho považovat za potenciální komunitu. Pokud Q_c je nula, tak propojení mezi N_c vrcholy je náhodné, plně založené na distribuci stupňů. A nakonec, pokud Q_c je záporné, tak vrcholy z C_c komunitu netvoří.

S použitím Rovnice (9) lze vyvodit jednodušší formu pro modularitu uvedenou v Rovnici (10).

$$Q_c = \frac{L_c}{L} - \left(\frac{k_c}{2L}\right)^2 \tag{10}$$

kde L_c je celkový počet hran v komunitě C_c a k_c je celkový stupeň vrcholů z téže komunity.

Abychom viděli, zda hustota podgrafů daného rozdělení se liší od očekávané hustoty v náhodně propojené síti, tak definujeme modularitu rozdělení, vypočtenou jako součet modularit všech n_c komunit (Rovnice (11).

$$Q = \sum_{c=1}^{n_c} \left[\frac{L_c}{L} - \left(\frac{k_c}{2L}\right)^2\right] \tag{11}$$

3.7.2 Louvain algoritmus

Tento algoritmus slouží pro rozdělení velké sítě na části s vysokou modularitou v krátkém čase a vytvoří kompletní hierarchii komunitní struktury pro vstupní síť a tím umožní výběr toho, jak bude síť rozdělena. Byl navržen Blondel; Guillaume; Lambiotte; Lefebvre [64] z Katolické univerzity ve městě Lovaň (francouzsky Louvain), ze kterého pochází název pro algoritmus.

Algoritmus je rozdělen na dvě části, které jsou iterativně opakovány. Vstupem je ohodnocená síť s N vrcholy. Na začátku je každý vrchol umístěn do samostatné komunity, takže na počátku je počet komunit roven N. Následně jsou pro každý vrchol i získáni jeho sousedé a vyhodnocuje se, zda se modularita zvýší odstraněním vrcholu i z jeho komunity a následným umístěním do komunity vrcholu j. Vrchol i je poté umístěn do komunity, ve které je nárůst modularity nejvyšší. Pokud k nárůstu modularity nedojde pro žádného ze sousedů, pak vrchol i zůstane ve své původní komunitě. Tento proces je opakován sekvenčně pro všechny vrcholy dokud dochází ke zlepšení modularity. Poté je první fáze hotova. Výpočet modularity pro vážený graf u tohoto algoritmu je definován v Rovnici (12)

$$Q_l = \frac{1}{2m} \sum_{i}^{N} \sum_{j}^{N} \left[A_{ij} \frac{k_i k_j}{2m}\right] \delta(c_i, c_j) \tag{12}$$

kde A_{ij} reprezentuje váhu hrany mezi vrcholy i a j, N je počet vrcholů sítě, $k_i = \sum_j^N A_{ij}$ je součtem všech vah hran incidentních s vrcholem i, c_i je komunita, do které vrchol i patří, funkce $\delta(c_i, c_j)$ je Kroneckerovo delta a $m = \frac{1}{2}\sum_i^N \sum_j^N A_{ij}$.

V druhé fázi algoritmu se vytváří nová síť, jejíž vrcholy jsou komunity nalezené v první fázi. Váhy hran mezi vrcholy v této nové síti jsou získány sečtením vah hran mezi vrcholy korespondujících dvou komunit. Hrany mezi vrcholy stejné komunity vedou ke smyčkám v nové síti. Po dokončení druhé fáze je možné opět aplikovat první fázi algoritmu na nově vzniklou ohodnocenou síť. Tyto dvě fáze jsou označovány jako „průchod“. Počet komunit se snižuje s každým průchodem. Průchody jsou opakovány dokud není dosažena maximální modularita.

Výhoda algoritmu je, že je velmi rychlý i pro rozsáhlé sítě s časovou složitostí $O(nlogn)$, která ovšem u řídkých grafů může být téměř lineární. Simulace ukázaly, že je to způsobeno tím, že nárůst modularity lze snadno spočítat a počet komunit značně klesá s každým průchodem, tedy nejvíce časově náročné jsou první průchody.

4 Vlastní implementace

Od této sekce práce přechází v praktickou část. Tato sekce se věnuje detailnímu popisu vlastní implementace systému pro analýzu vícevariačních sítí, zahrnujícímu veškeré náležitosti softwarového inženýrství. Sekce je dále rozdělena na dokument specifikace požadavků, obsahující detailní popis funkcí systému, popis architektury a část experimentální, ve které jsou uvedeny výsledky experimentů pro vybrané testovací vícevariační sítě.

4.1 Specifikace požadavků

Tato sekce obsahuje dokument specifikace požadavků na systém pro analýzu vícevaričních sítí. Tento dokument specifikace požadavků obsahuje popis funkcí a požadavků specifikovaných pro tento systém, detailní popis architektury systému včetně jeho závislostí na externích knihovnách a popis uživatelů, kteří budou se systémem pracovat. Uvedeny zde budou také omezení, která pro tento systém existují. Závěr této sekce tvoří seznam požadavků s jejich podrobným rozborem.

4.1.1 Účel systému

Průzkum, analýza a porozumění komplexním a rozsáhlým datovým sadám je náročnou výzvou. Nalezení možností, jak tyto vztahy rozlišit v kontextu dalších souvisejících dat, je velmi důležité pro řadu různých vědeckých oblastí. Systém pro analýzu více variačních sítí je tedy určen pro vytvoření vizualizace vícevariačních sítí. Vizualizace je využívána pro náhled a porozumění mezi komplexními objekty a jejich vlastnostmi. Při vizualizaci vícevariačních sítí se objevuje mnoho výzev, mezi než patří hlavně způsob, jak zvýšit srozumitelnost zobrazených dat pro uživatele a další problémy týkající se rozsáhlých, komplexních a dynamických dat. Systém je navržen tak, aby si s těmito problémy dokázal co nejlépe poradit. Implementované metody umožňují uživateli upravit si rozložení a vzhled vizualizace sítě dle jeho preferencí, a tím mu umožňuje vytvořit přehlednou vizualizaci, ze které poté může čerpat potřebné informace o datové sadě.

4.1.2 Funkce systému

Systém pro analýzu vícevariačních sítí uživateli umožňuje provést vizualizaci vícevariační sítě, kterou si dále může prozkoumávat a upravovat dle svých potřeb. Systém umožňuje uživateli nahrát soubor, obsahující vektorová data, která splňují předepsaný formát. Dále pokud uživatel vlastní i soubor s korespondující sítí, je mu dána možnost tento soubor nahrát také. V případě, že síť není uživatelem dodána, tak je systém schopen využít vektorová data k převodu na síť. Takto načtená vícevariační síť je systémem vizualizována a prezentována uživateli. Systém umožňuje vytvořenou vizualizaci modifikovat více způsoby. Uživatel je schopen nastavit meze jednotlivých atributů vrcholů a odstranit ty, které svou hodnotou do těchto mezí nespadají. Rozložení sítě není pevné, proto uživatel může vlastnosti grafu sítě modifikovat dle své libosti. Další funkcí, kterou lze využít, je rozřazování vrcholů grafu do skupin. Skupiny lze tvořit ručně nebo lze využít

funkce detekce komunit, při které systém samostatně rozdělí vrcholy do komunit, tak aby byla dosažena co nejvyšší hodnota jejich modularity.

V Tabulce 5 je uveden kompletní přehled funkčních požadavků. Na Obrázku 8 je zobrazen diagram případů užití, který ukazuje přehled případu užití, které mohou nastat při práci se systémem. Tyto případy odpovídají funkčním požadavkům kladeným na systém.

ID	Název	Specifikace	Sekce
1	Vstupní data uživatele	Systém musí umožňovat zadaní cesty k souborům vícevariační sítě a parametry pro její vizualizaci.	4.1.3
2	Načtení vektorových dat ze souboru	Systém musí umět načíst vektorová data ze souboru do datové struktury v pracovní paměti.	4.1.4
3	Načtení sítě ze souboru	Systém musí umět načíst síť ze souboru do datové struktury v pracovní paměti.	4.1.5
4	Převod vektorových dat na síť	Systém musí umět zkonstruovat síť z daných vektorových dat .	4.1.6
5	Vizualizace vícevariační sítě	Systém musí umět vytvořit grafickou reprezentaci vícevariační sítě.	4.1.7
6	Zobrazení vektorových dat	Systém musí umět uživateli zobrazit data o daném vrcholu vícevariační sítě.	4.1.8
7	Detekce komunit	Systém musí umět detekovat komunity v načtené vícevariační síti.	4.1.9
8	Tvorba a úprava skupin vrcholů	Systém musí umožňovat uživateli vytvářet skupiny pro rozřazování vrcholů.	4.1.10
9	Vizualizace sítě skupin	Systém musí umět vizualizovat síť reprezentující vztahy mezi vytvořeními skupinami.	4.1.11
10	Filtrace vrcholů	Systém musí umět filtrovat vrcholy na základě daných hodnot atributů.	4.1.12

Tabulka 5: Přehled funkčních požadavků systému pro analýzu vícevariačních sítí

4.1.3 Požadavek č. 1: Vstupní data uživatele

Systém musí umožňovat zadaní cesty k souborům vícevariační sítě a parametry pro její vizualizaci, což je počátkem celého procesu vizualizace vícevariační sítě.

Uživatel na domovské stránce systému vloží cestu k souboru s vektorovými daty. Cesta k tomuto souboru je jedna ze tří povinných vstupních hodnot potřebných k procesu vizualizace vícevariační sítě. Cestu může uživatel zadat pomocí okna pro prohledávání souborů.

Uživatel může dále zadat cestu k souboru obsahujícímu data o struktuře sítě. Zadání této cesty není povinné. Pokud není cesta k souboru určena, tak se síť vytvoří automaticky na základě vstupních vektorových dat.

Prvním parametrem je rozhodnutí, zda zadané soubory s daty o vícevariační síti obsahují jména atributů či ne. Tento parametr je druhou povinnou vstupní hodnotou. Pokud uživatel nezadá tuto správně, tak dojde k chybě při procesu vizualizace, jelikož data nebudou správně přečtena. Přítomnost názvů atributů ve vstupních souborech může uživatel určit pomocí zaškrtávacího pole, které ve výchozím stavu není zaškrtnuto.

Druhým parametrem jsou oddělovací znaky. Uživatel zadá do vstupního pole znaky, kterými jsou data v souboru oddělena. Tento parametr je povinný. Jednotlivé znaky ve vstupním poli jsou při zadávání odděleny mezerou. Pokud vstupní pole znaků není vyplněno, tak se jako oddělovací znak použije mezera. Systém nedovoluje jako oddělovací znak více než jednu mezeru.

Třetím parametrem je volba algoritmu pro převod vektorových dat na síť v případě, že uživatel nevložil cestu k souboru se strukturou sítě. Algoritmus může uživatel vybrat pomocí přepínacích tlačítek, kde jako výchozí algoritmus je zvolen algoritmus *LRNet* 3.4.2. Pokud je zvolen algoritmus pro kontrukci ϵ a kNN grafu, pak uživatel musí také zadat práh podobnosti vrcholů a minimální počet sousedů vrcholu pro tento algoritmus.

Čtvrtým parametrem je rozhodnutí o tom, zda je vícevariační síť orientovaná nebo ne. Orientaci sítě může uživatel určit pomocí zaškrtávacího pole, které ve výchozím stavu není zaškrtnuto.

Pátým parametrem je jméno atributu, který v datové sadě reprezentuje identifikační čísla jednotlivých záznamů. Tento parametr je volitelný a umožní systému přiřadit reálná jména vrcholům. Pokud se atribut nachází v datové sadě a není zadáno jeho jméno, tak je tento atribut zpracován obecným způsobem.

Šestým parametrem je jméno atributu, který v datové sadě reprezentuje reálné třídy jednotlivých záznamů. Tento parametr je volitelný a umožní systému obarvit vrcholy dle jejich reálných komunit. Pokud se atribut nachází v datové sadě a není zadáno jeho jméno, tak je tento atribut zpracován obecným způsobem.

Posledním parametrem je rozhodnutí, zda mají být vrcholy vícevariační sítě rozřazeny do komunit již při počátečním zobrazení její vizualizace. Rozhodnutí, zda se má detekce komunit provést, může uživatel určit pomocí zaškrtávacího pole, které ve výchozím stavu není zaškrtnuto.

Při zadávání vstupních dat uživatelem je systém nečinný. Jakmile uživatel klikne na tlačítko „Upload“, tak jsou soubory s parametry odeslány na server přes TCP/IP síť. Server poté zahájí načtení vektorových dat do pracovní paměti. Systém následně pokračuje převodem vektorových dat na síť nebo načtením sítě ze souboru. V získané síti poté může systém provést proces detekce komunit. Všechny tyto procesy jsou popsány níže. Pokud dojde při některém z těchto procesů k chybě, tak je uživatel vrácen zpět na domovskou stránku s upozorněním na danou chybu.

Data v souborech musí splňovat určitý formát. Jednotlivé hodnoty atributů či vrcholy sítě musí být odděleny pomocí oddělovacích znaků. Soubory kromě těchto oddělených dat můžou obsahovat pouze názvy atributů a to na začátku souboru. Data mohou obsahovat chybějící hodnoty, ovšem tyto chybějící hodnoty se nesmí vyskytovat v prvním záznamu datové sady. Pokud data v souborech nedodrží tento formát, tak bude uživatel vrácen zpět na domovskou stránku s chybovou zprávou. V souboru se strukturou sítě musí řádek obsahovat pouze dvojici

vrcholů mezi nimiž je hrana. Tyto vrcholy jsou odděleny znakem. Počet vrcholů v síti musí odpovídat počtu záznamů ve vektorových datech.

4.1.4 Požadavek č. 2: Načtení vektorových dat ze souboru

Systém musí umět načíst vektorová data ze souboru do datové struktury v pracovní paměti. Vstupem pro tuto funkci je uživatelem odeslaný soubor na server, který obsahuje vektorová data. Na vstupu také ještě dva parametry. Prvním parametrem jsou znaky, pomocí kterých jsou data v souboru oddělena a druhým je rozhodnutí o tom, zda soubor obsahuje na začátku názvy atributů. Systém tento soubor otevře a vytvoří datovou strukturu s počtem a názvy atributů v souboru. Dále jsou vektorová data postupně ukládána do této datové struktury. Výsledná datová struktura obsahující všechna vektorová data je poté předána do dalších procesů.

4.1.5 Požadavek č. 3: Načtení sítě ze souboru

Systém musí umět načíst síť ze souboru do datové struktury v pracovní paměti. Vstupem pro tuto funkci je uživatelem odeslaný soubor na server, který obsahuje data o struktuře sítě. Na vstupu také ještě dva parametry. Prvním parametrem jsou znaky, pomocí kterých jsou data v souboru oddělena a druhým je rozhodnutí o tom, zda výsledná síť má být orientovaná. Systém tento soubor otevře a vytvoří datovou strukturu pro uchování sítě v paměti. Dále jsou jednotlivé hrany a vrcholy sítě postupně ukládány do této datové struktury. Výsledná datová struktura obsahující strukturu sítě je poté předána do dalších procesů.

4.1.6 Požadavek č. 4: Převod vektorových dat na síť

Systém musí umět zkonstruovat síť z daných vektorových dat. Vstupem pro tuto funkci je datová struktura s vektorovými daty a vybraný algoritmus pro převod. Na základě vlastností nalezených v těchto vektorových datech algoritmus zkonstruuje síť a uloží ji do datové struktury. Výsledná datová struktura obsahující strukturu sítě je poté předána do dalších procesů.

4.1.7 Požadavek č. 5: Vizualizace vícevariační sítě

Systém musí umět vytvořit grafickou reprezentaci vícevariační sítě. Vstupem pro tuto funkci je datová struktura obsahující síťová data. Systém uživateli načte webovou stránku, která bude obsahovat komponentu, ve které se okamžitě vytvoří graf reprezentují vstupní síť. Mimo tohoto grafu bude na stránce umístěna komponenta se seznamem všech vrcholů, kde u každého vrcholu budou uvedeny jeho atributy s příslušnými hodnotami.

4.1.8 Požadavek č. 6: Zobrazení vektorových dat

Systém musí umět uživateli zobrazit data o daném vrcholu vícevariační sítě. Vstupem pro tuto funkci je datová struktura, obsahující vektorová data a vybraný vrchol vícevariační sítě. Webová

stránka, ve které je vykreslen graf vícevariační sítě, musí dále obsahovat seznam veškerých vrcholů, u kterých je možno nahlédnout na jejich hodnoty atributů. Uživatel může vrchol, jehož detaily ho zajímají, nalézt v seznamu vrcholů a nebo tento vrchol označí kliknutím přímo v grafu vícevariační sítě a systém mu poté detaily o vrcholu zobrazí.

4.1.9 Požadavek č.7: Detekce komunit

Systém musí umět detekovat komunity v načtené vícevariační síti. Vstupem pro tuto funkci je datová struktura, obsahující síťová data. Uživatel si před vizualizací vícevariační sítě a nebo poté kdykoliv při interaktivním průzkumu vícevariční sítě může nechat systém detekovat komunity v síti. Detekce komunit je uskutečněna pomocí Louvain algoritmu (Sekce 3.7.2). Vrcholy jsou poté v grafu obarveny dle příslušné komunity a následně je vytvořen seznam těchto komunit, které je možno dále upravovat.

4.1.10 Požadavek č.8: Tvorba a úprava skupin vrcholů

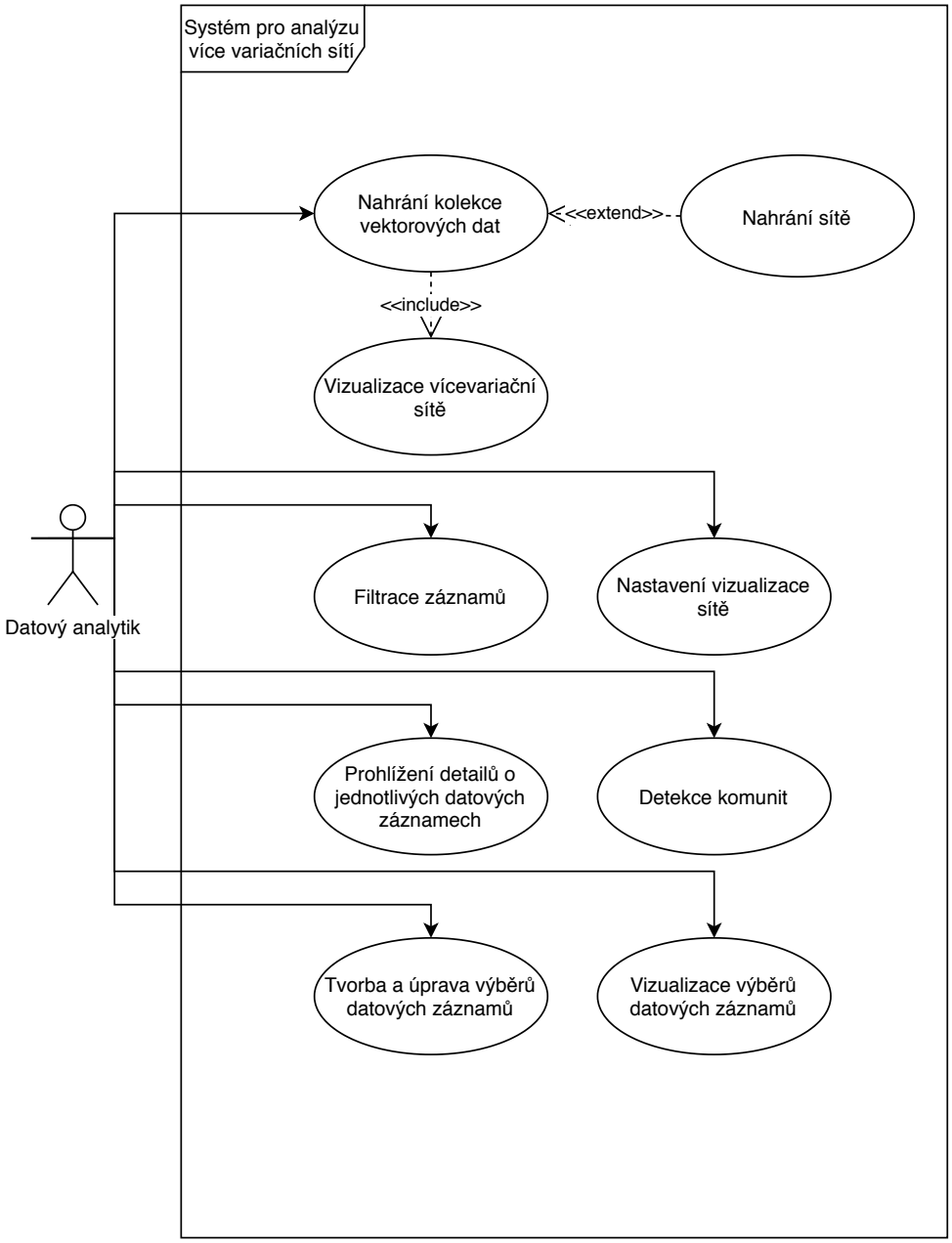
Systém musí umožňovat uživateli vytvářet skupiny pro rozřazování vrcholů. Vstupem pro tuto funkci je datová struktura, obsahující síťová data a výběr vrcholů. Uživatel si kdykoliv při interaktivním průzkumu vícevariční sítě může upravovat skupiny, do kterých může poté přidávat nebo z nich odebírat vrcholy sítě. Vrcholy jsou poté v grafu obarveny dle příslušné skupiny a je evidován seznam těchto skupin.

4.1.11 Požadavek č.9: Vizualizace sítě skupin

Systém musí umět vizualizovat síť reprezentující vztahy mezi vytvořeními skupinami. Vstupem pro tuto funkci je seznam komunit či skupin, kde každá komunita či skupina obsahuje seznam vrcholů, které se v ní nacházejí. Po detekci komunit či jakékoliv úpravě skupin bude kromě grafu sítě vytvořena vizuální reprezentace vztahů mezi jednotlivými skupina či komunitami. K tomuto bude opět využit graf, kde vrcholy reprezentují jednotlivé komunity či skupiny a hrany reprezentují výskyt vztahu mezi nimi. Velikost vrcholu skupiny či komunity se bude zvyšovat s počtem vrcholů vícevariční sítě, které obsahuje. U hran bude číslem uvedeno, kolik hran mezi příslušnými skupinami či komunitami existuje.

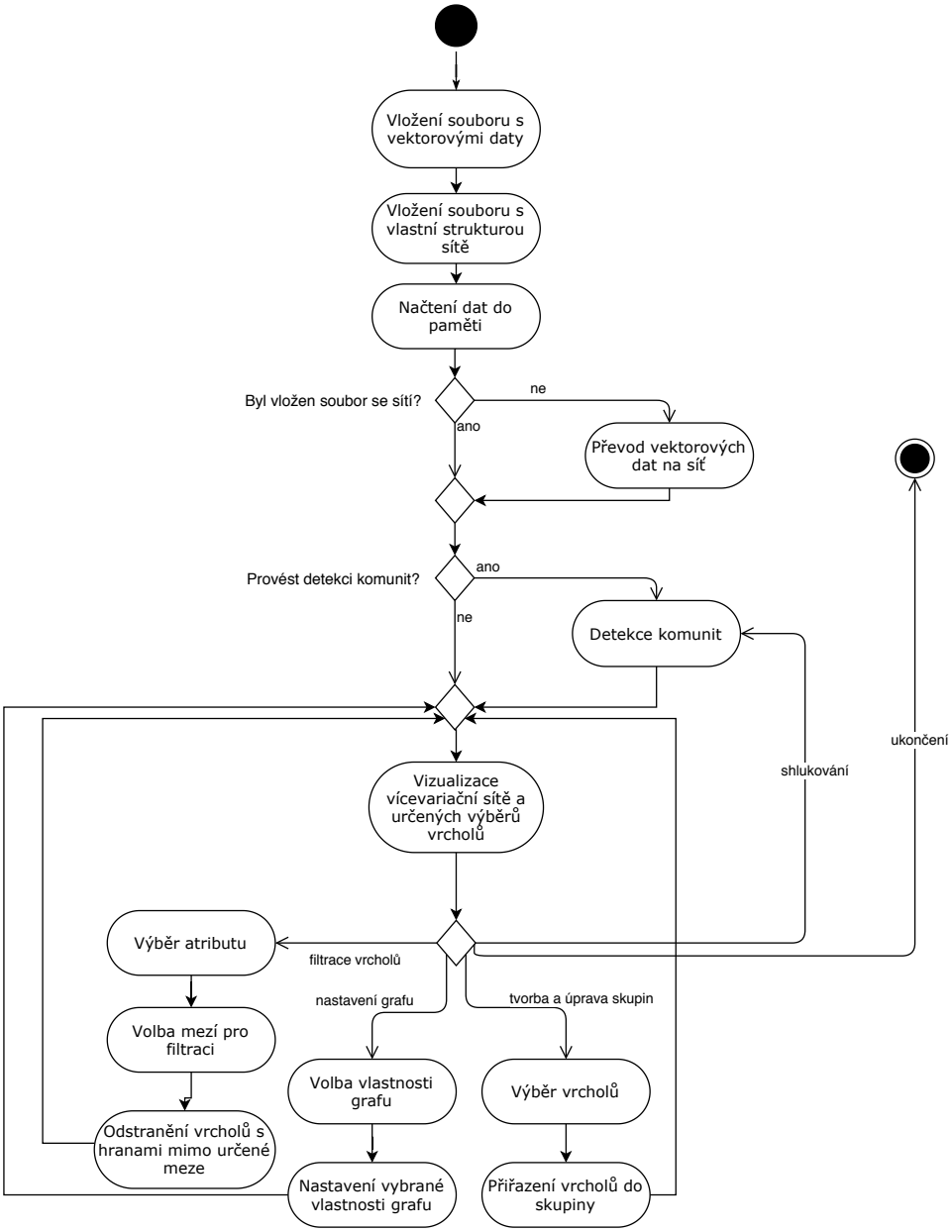
4.1.12 Požadavek č.10: Filtrace vrcholů

Systém musí umět filtrovat vrcholy na základě daných hodnot atributů. Vstupem pro tuto funkci je vybraný atribut a jeho meze. Pro numerické atributy uživatel zadává jejich minimální a maximální povolenou hodnotu a u kategoriálních si pouze zvolí, které kategorie chce zachovat a které ne. Po nastavení filtračních hodnot atributů budou vrcholy nesplňující zadaná kritéria odstraněny z grafu vícevariační sítě.



Obrázek 8: Diagram případu užití systému pro analýzu vícevariačních sítí

Na Obrázku 9 je zobrazen postup práce se systémem pro analýzu vícevariačních sítí. Systém nejdříve vyzve uživatele k zadání cesty k souboru s vektorovými daty. Uživatel může nepovinně vložit i cestu k souboru obsahující data o korespondující síti. Uživatel si ještě před samotným nahráním souborů do systému může zvolit, zda chce rovnou využít možnosti detekce komunit k rozdělení vrcholů do skupin. Po vytvoření vizualizace zadané sítě a její následné prezentaci si uživatel může opakovaně vizualizaci upravovat a prozkoumávat k získání potřebných dat o síti.



Obrázek 9: Diagram aktivit pracovního postupu se systémem pro analýzu vícevariačních sítí

4.1.13 Technické požadavky

Tato sekce obsahuje výpis požadavků, které jsou vyžadovány pro správný běh systému. Zahrnuje požadavky kladené na uživatele a systémové požadavky.

4.1.13.1 Požadavky na uživatele U uživatele se předpokládá znalost problematiky vícevariačních sítí, detekce komunit a znalost principu algoritmů implementovaných v systému. Uživatel se dále musí ujistit, že vstupní soubory obsahují data, dodržující formát popsaný v Sekci 4.1.3. U uživatele se očekává alespoň základní znalost práce s počítačem, zahrnující hlavně práci s

textovými editory pro úpravu dat. Celý systém je lokalizován v anglickém jazyce bez možnosti změny jazyka. Uživatel by tedy měl mít znalost angličtiny na středně pokročilé úrovni.

4.1.13.2 Systémové požadavky Systém byl vyvíjen ve vývojovém prostředí Microsoft Visual Studio 2019 jako webová aplikace na platformě ASP.NET Core. Hostující server musí tuto platformu podporovat. Tato platforma je volně dostupná a běží na operačních systémech Windows, Linux a macOS s procesory x64, x86, ARM32 a ARM 64. Jako programovací jazyk pro aplikaci na straně serveru byl zvolen C# s jazykem HTML a JavaScript na straně klientské.

Počítač, na kterém byl systém vyvíjen a testován, měl nainstalován operační systém Windows 10 64 bit s procesorem Intel Core i7-9750H a operační pamětí 16 GB. Aplikace by měla běžet ve všech moderních prohlížečích. Jako testovací prohlížeč byl použit prohlížeč Mozilla Firefox. Při používání aplikace je doporučeno používat stejné technologie a aplikace jako při testování k minimalizaci neočekávaných chyb.

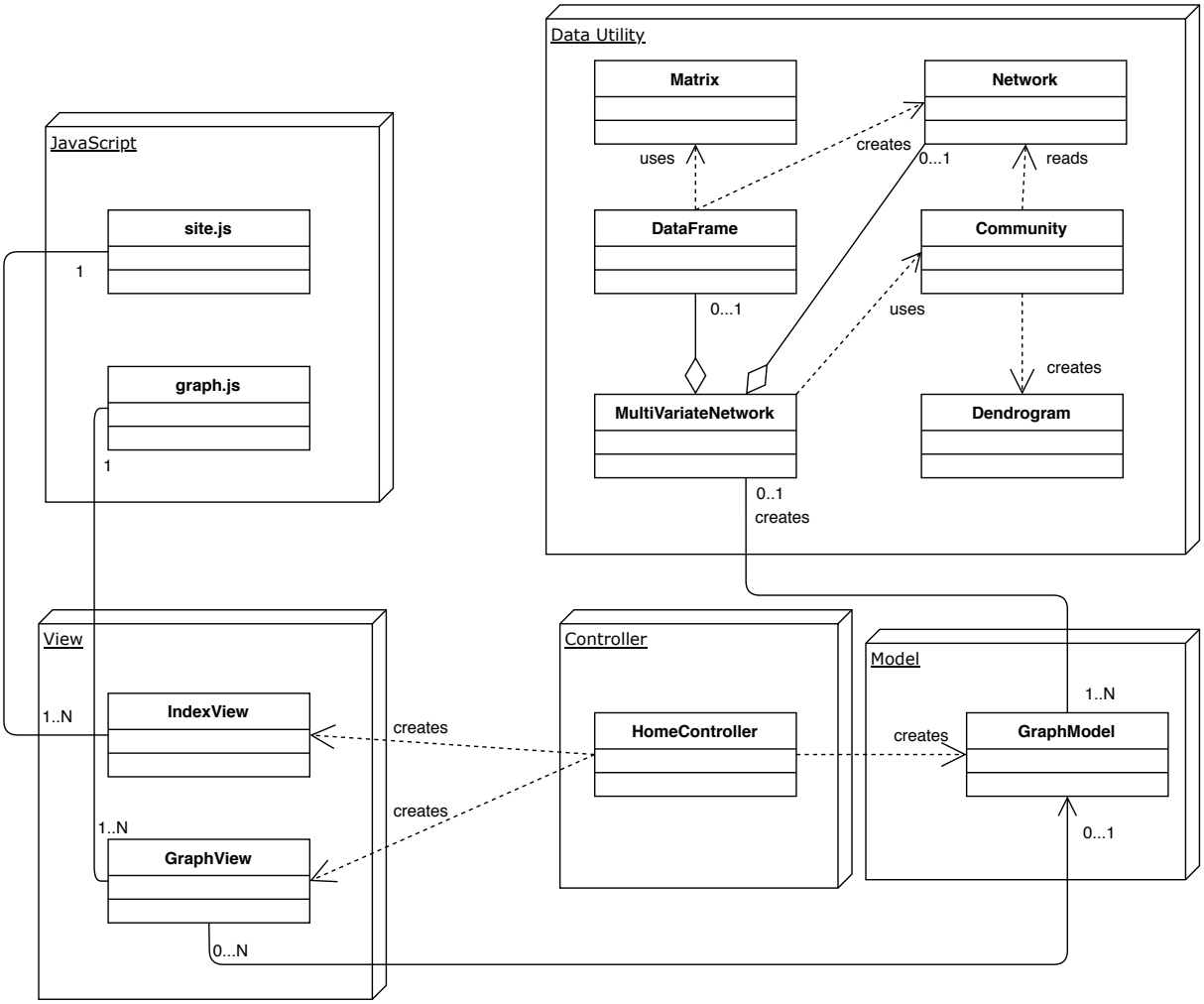
4.2 Přehled vývoje a architektury

Systém je strukturován pomocí architektury klient-server a pro oddělení aplikační logiky od prezentace uživateli využívá architektonický vzor Model-Pohled-Kontrolér (MVC). Systém je dále závislý na vlastní knihovně, která obsahuje datové struktury a veškerou logiku pro práci s vícevariačními sítěmi. Dále je systém závislý na dvou dalších externích knihovnách.

Struktura systému je vyobrazena na diagramu tříd na Obrázku 10. Jelikož je diagram využit ke znázornění struktury systému, tak třídy neobsahují žádná datová pole ani metody. Navíc jsou zde zahrnuty i pohledy z architektury MVC a jejich javascriptové soubory, které jsou v tomto digramu také považovány za třídy, která nabízejí data či metody.

Data-driven Documents (D3) [65] je volně dostupná JavaScriptová knihovna sloužící pro manipulaci dokumentů na základě dat. D3 přivádí data k životu za pomocí HTML, SVG a CSS. Zaměření knihovny D3 na webové standarty dává uživateli plnou kapacitu moderních webových prohlížečů bez závislosti na proprietárních rámcích. Tato knihovna kombinuje vizualizační komponenty a datově řízený přístup k manipulaci s objektovým modelem dokumentu (DOM). V systému pro analýzu vícevariačních sítí je tato knihovna primárně využívána pro vytvoření grafu sítě.

Json.NET [66] je populární volně dostupný JSON rámec pro .NET, který se využívá pro tvorbu a úpravu JSON objektů. Knihovna nabízí také třídy pro serializaci a deserializaci jakýkoliv objektů v .NETu. V systému je tato knihovna využita pro převod vícevariační sítě do JSON objektu, který je následně využit na straně klienta pro tvorbu grafu sítě za pomocí knihovny D3.



Obrázek 10: Diagram komponent systému pro analýzu vícevariačních sítí

4.3 Experimenty

V této sekci jsou popsány experimenty a průzkum vícevariačních dat pomocí systému pro analýzu vícevariačních sítí. Cílem je ukázat, zda Louvain algoritmus (3.7.2) pro detekci komunit rozdělí vrcholy sítě, kterou získáme převodem z vektorových dat, do komunit, které budou co nejvíce odpovídat jejich reálným třídám. Pokud ne, tak jestli je možné tuto detekci komunit zlepšit pomocí filtrování vrcholů na základě jejich atributů. Experimenty byly provedeny na třech testovacích datových sadách, obsahujících vícevariační data. Jmenovitě jde o datové sady *Ecoli*, *Mice protein Expression* a *Audit Data*. Všechny tyto sady byly získány z repozitáře strojového učení UCI [67].

Pro určení kvality převodu vektorových dat na síť je využita klasifikační přesnost, která popisuje, do jaké míry jsou propojeny vrcholy patřící do stejných reálných tříd [49]. První takovou přesností je průměrná vážená přesnost, což je průměrná hodnota výpočtu v Rovnici (13), který se provádí pro všechny vrcholy, kde ω_{pos} je součet vah hran se sousedy ze stejné třídě a ω_{all} je součet všech vah hran se všemi sousedy. Další přesnost je průměrná přesnost, která je průměrem hodnoty p všech vrcholů. Hodnota $p = 1$ pro zvolený vrchol, pokud součet všech vah hran se sousedy ze stejné třídy je vyšší než součet vah hran se sousedy z ostatních tříd.

$$\omega = \frac{\omega_{pos}}{\omega_{all}} \tag{13}$$

Pro určení kvality shlukování byl využit Silhouette koeficient pro jednotlivé vrcholy [68], support (podpora), který je uveden v Rovnici (14), kde r je počet vrcholů ve shluku a N je počet všech vrcholů grafu a confidence (spolehlivost), uvedený v Rovnici (15), kde c_{max} je maximální počet vrcholů jedné zastoupené třídy v komunitě.

$$support = \frac{r}{N} \tag{14}$$

$$confidence = \frac{c_{max}}{r} \tag{15}$$

4.3.1 Ecoli

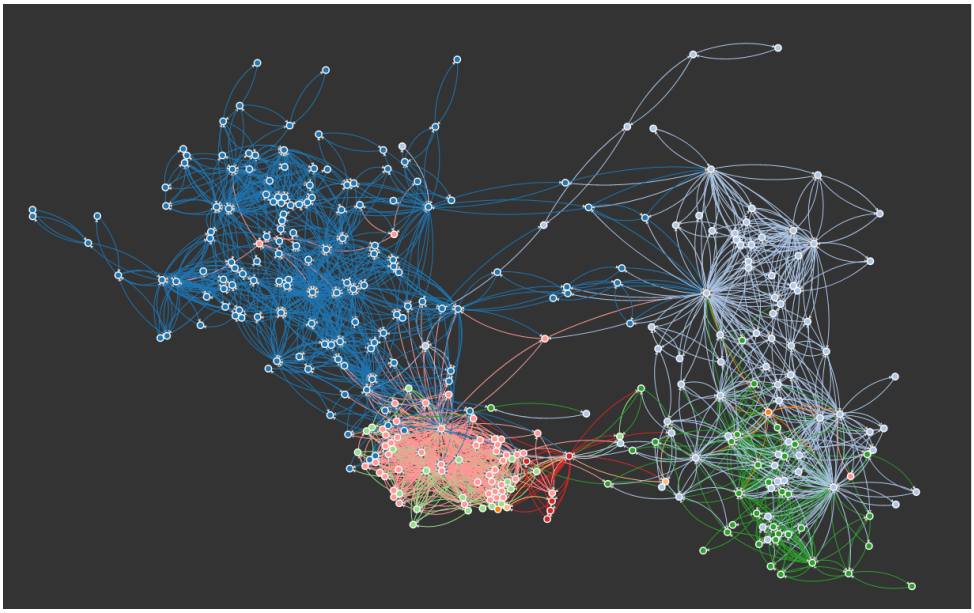
Datová sada Ecoli obsahuje data o proteinech, které slouží pro klasifikaci. Tato datová sada obsahuje 336 datových záznamů s vícevariačními daty. Každý záznam má 1 kategoriální atribut, označující název proteinu a 7 numerických atributů. Data jsou rozdělena do osmi tříd, určujících místo výskytu proteinu.

Vícevariační data jsou po jejich načtení převedena na síť pomocí algoritmů LRNet a ϵ-kNN, u kterého byl práh podobnosti nastaven na 0,9998 a minimální počet sousedů na 1. Vzniklé grafy sítí jsou neorientované, ovšem neorientované hrany jsou reprezentovány dvěma orientovanými hranami pro lepší viditelnost hran mezi komunitami. Na všech obrázcích jsou vrcholy rozmístěny dle hodnot atributu *alm1* na ose X a dle hodnot atributu *mcg* na ose Y. Osa Y je zobrazena

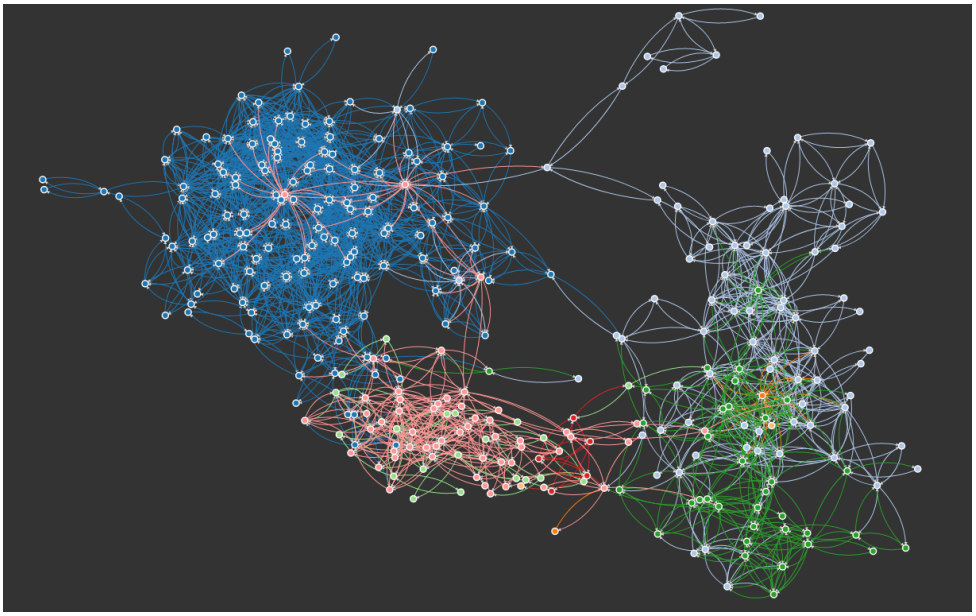
obráceně s menšími hodnotami nahoře a většími dole. Tyto atributy poskytly rozložení, které umožňuje nejlépe vidět komunitní strukturu sítě. Vrcholy mají stejnou barvu a jsou umístěny blíže k sobě, pokud patří do stejné komunity. V Tabulce 6 je uvedena klasifikační přesnost zkonstruovaných grafů pomocí algoritmů LRNet a ϵ-kNN. Na Obrázku 11 je zobrazena výsledná síť LRNet Ecoli a na Obrázku 12 síť ϵ-kNN Ecoli. Na obou obrázcích jsou vrcholy rozděleny do komunit dle reálných tříd. Na Obrázku 13 je zobrazen sloupcový graf silhouette koeficientů pro jednotlivé vrcholy, rozdělené dle reálných tříd. Dle tohoto grafu lze usoudit, že vrcholy nejde jednoznačně přiřadit do tříd.

Graf	Vážená přesnost	Přesnost
Ecoli: LRNet	0.798	0.804
Ecoli: ϵ-kNN	0.806	0.821

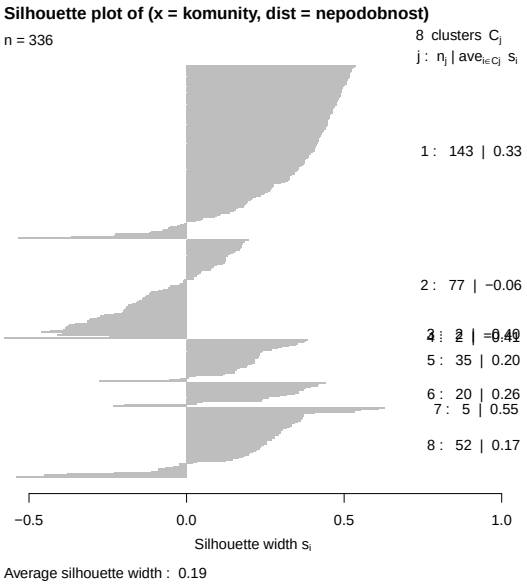
Tabulka 6: Klasifikační přesnost zkonstruovaných grafů



Obrázek 11: LRNet Ecoli síť s reálnými třídami vrcholů

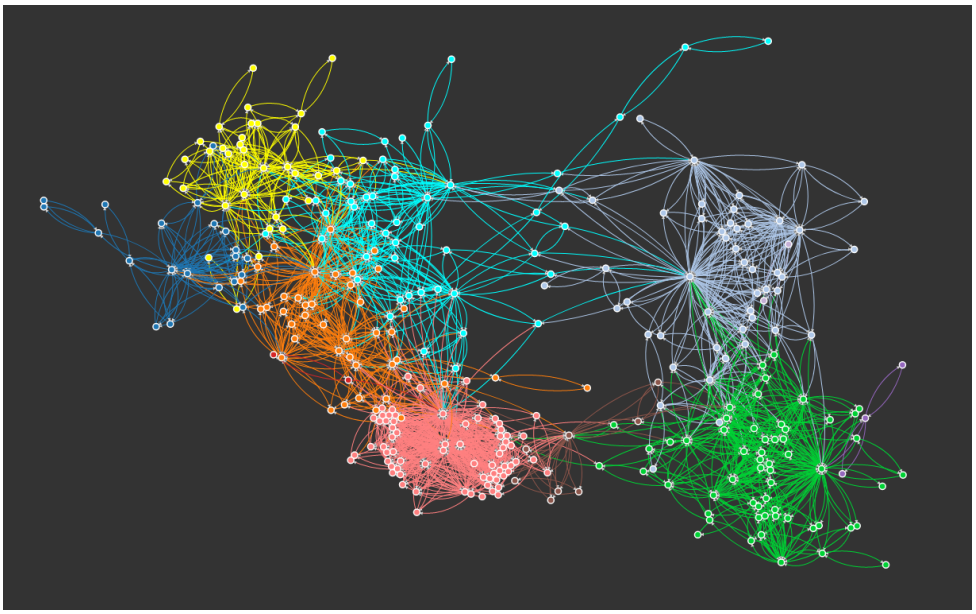


Obrázek 12: ϵ-kNN Ecoli síť s reálnými třídami vrcholů

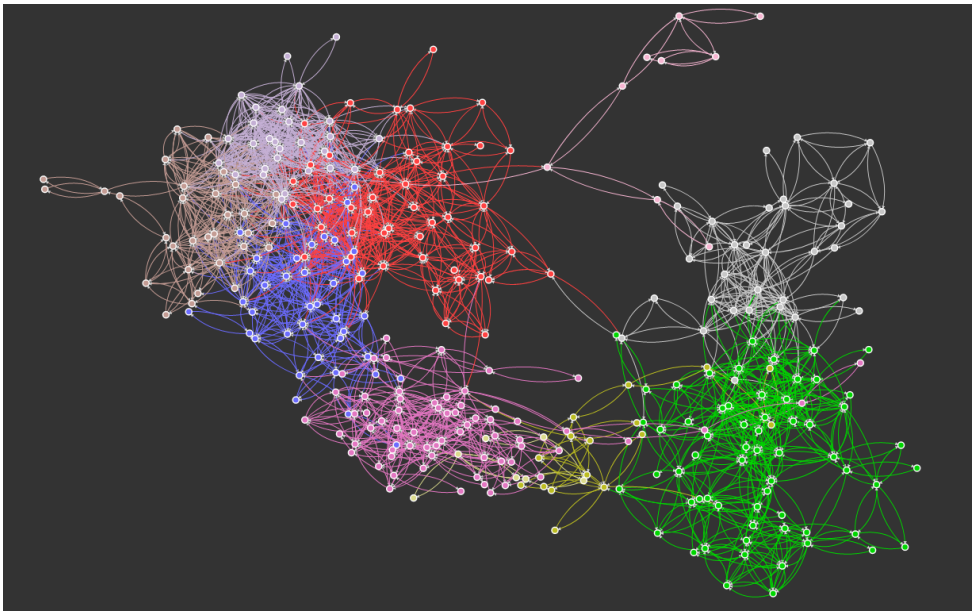


Obrázek 13: Silhouette koeficienty vrcholů rozdělených dle reálných tříd sítě Ecoli

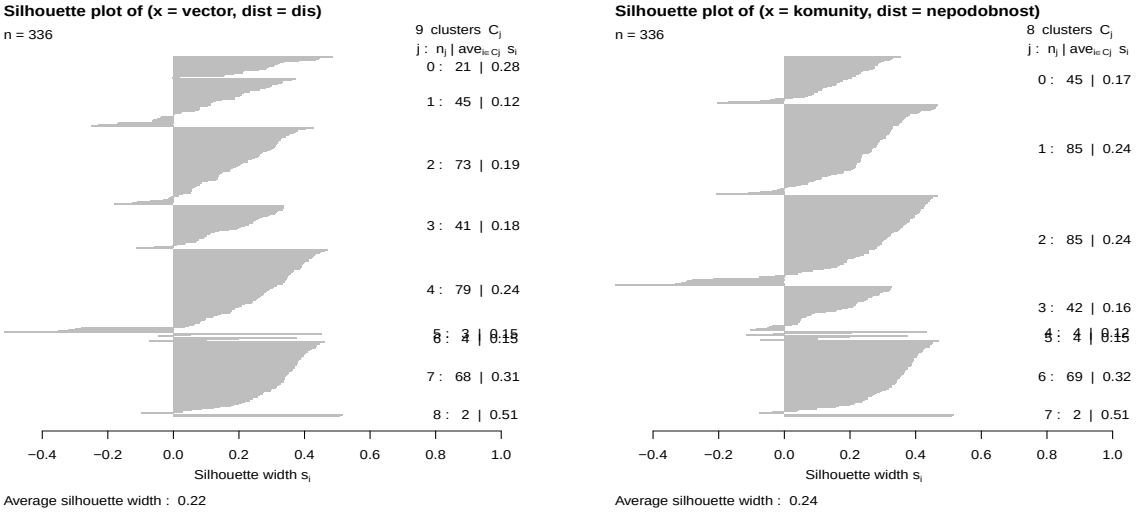
Na Obrázcích 14 a 15 je zobrazena celá síť Ecoli po detekci komunit pomocí Louvain algoritmu. Na Obrázku 16 je zobrazen sloupcový graf silhouette koeficientů pro jednotlivé vrcholy sítě Ecoli po detekci komunit. V Tabulce 7 jsou uvedeny support a confidence hodnoty pro jednotlivé nalezené komunity na celé síti Ecoli.



Obrázek 14: LRNet Ecoli síť s detekovanými komunitami



Obrázek 15: ϵ-kNN Ecoli síť s detekovanými komunitami



(a) LRNet Ecoli (b) ϵ-kNN Ecoli

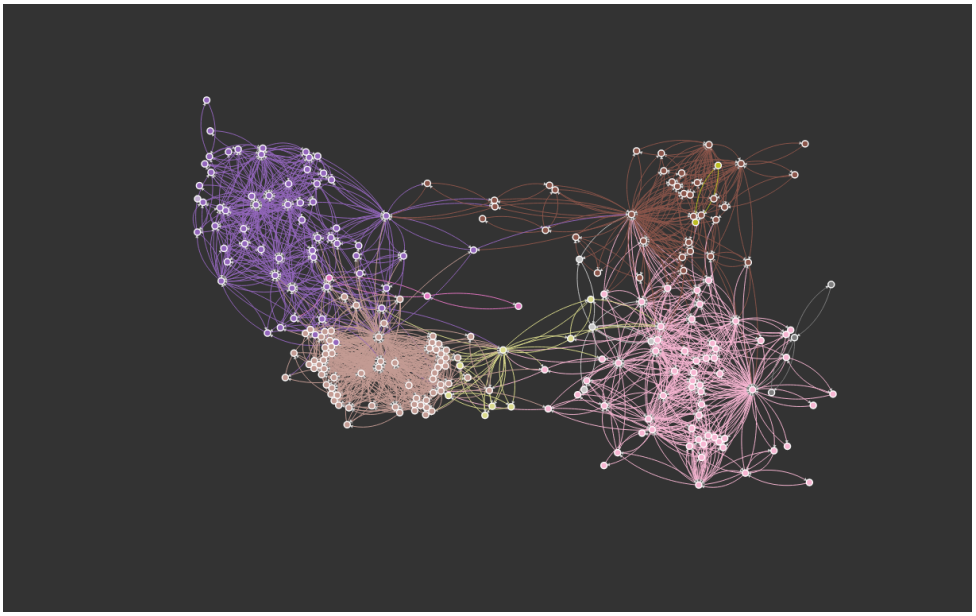
Obrázek 16: Silhouette koeficienty vrcholů detekovaných komunit na celé síti Ecoli

Support LRNet Ecoli	Confidence LRNet Ecoli	Support ϵ-kNN Ecoli	Confidence ϵ-kNN Ecoli
0,217	0,644	0,190	0,703
0,199	0,493	0,184	0,500
0,154	0,923	0,145	0,980
0,122	0,854	0,116	1,000
0,107	1,000	0,104	0,886
0,092	0,871	0,098	0,939
0,062	1,000	0,056	1,000
0,017	0,833	0,047	0,313
0,011	1,000	0,026	1,000
0,011	1,000	0,020	1,000
0,008	0,667	0,008	1,000

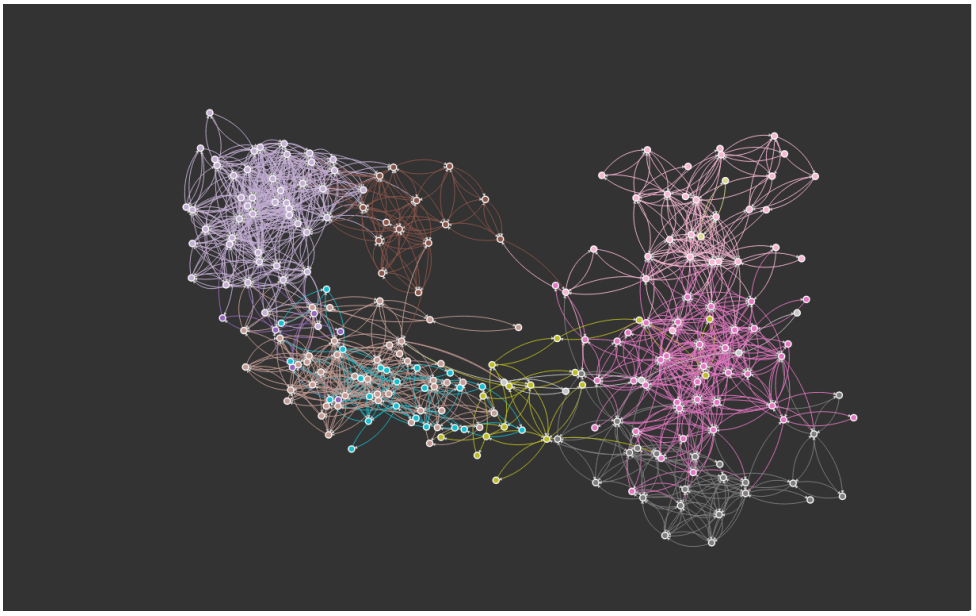
Tabulka 7: Support a confidence detekovaných komunit na celé síti Ecoli

Je očividné, že komunity nacházející se na pravé straně přibližně odpovídají reálným třídám a nebyl nalezen atribut, podle kterého by bylo možné filtrovat vrcholy pro zlepšení rozdělení do komunit pro tuto část sítě. Vrcholy nacházející se na levé straně byly rozděleny do více komunit než jak je dáno reálnými třídami. Jako nejvhodnější atributy pro filtraci byly vybrány *alm1* a *mcg*, dle kterých je graf rozložen. Atribut *alm1* má rozsah od 0,03 do 1,00 a jeho spodní hranice byla omezena na 0,26. Atribut *mcg* má rozsah od 0,00 do 0,89 a jeho spodní hranice byla omezena na 0,31. Obě tato omezení způsobila odfiltrování 76 vrcholů. Filtrované sítě jsou zobrazeny na Obrázcích 17 a 18. Na Obrázku 19 je zobrazen sloupcový graf silhouette koeficientů jednotlivých

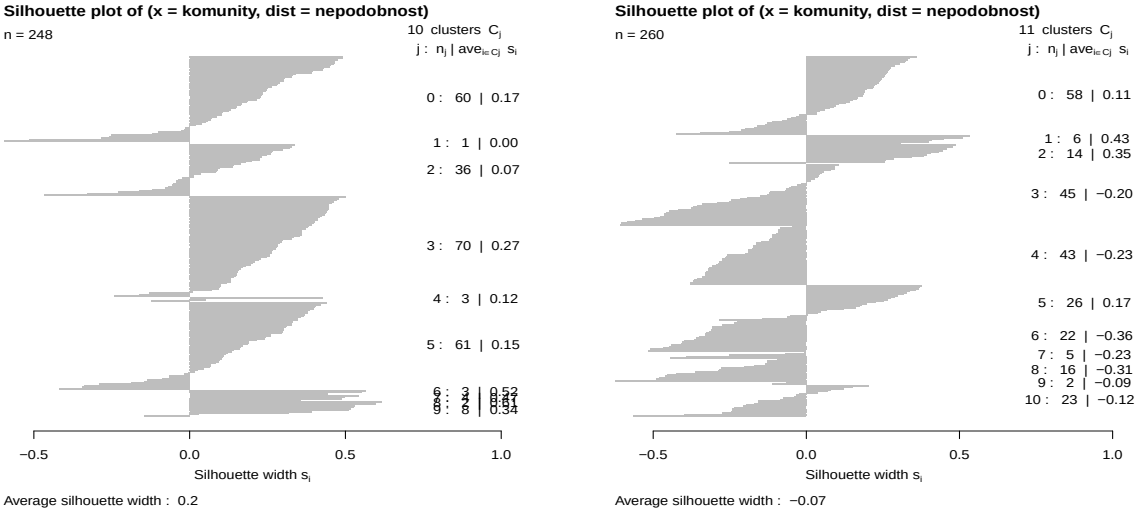
vrcholů filtrované sítě a v Tabulce 8 je uveden support a confidence pro jednotlivé komunity filtrované sítě.



Obrázek 17: Filtrovaná LRNet Ecoli síť s detekovanými komunitami



Obrázek 18: Filtrovaná ϵ-kNN Ecoli síť s detekovanými komunitami



(a) LRNet Ecoli (b) ϵ-kNN Ecoli

Obrázek 19: Silhouette koeficienty vrcholů detekovaných komunit na filtrované síti Ecoli

Support LRNet Ecoli	Confidence LRNet Ecoli	Support ϵ-kNN Ecoli	Confidence ϵ-kNN Ecoli
0,282	0,671	0,189	0,979
0,245	0,525	0,177	0,818
0,241	0,933	0,157	0,641
0,145	0,778	0,116	0,931
0,032	0,625	0,092	0,783
0,016	1,000	0,092	0,739
0,012	0,333	0,064	0,313
0,012	0,667	0,056	0,857
0,008	1,000	0,024	1,000
0,007	1,000	0,020	0,600
-	-	0,013	1,000

Tabulka 8: Support a confidence detekovaných komunit na filtrované síti Ecoli

4.3.2 Mice Protein Expression

Datová sada Mice Protein Expression obsahuje data o 77 úrovních proteinových modifikací u myší a slouží pro klasifikaci. Tato datová sada obsahuje 1080 datových záznamů s vícevariačními daty. Každý záznam má 1 kategoriální atribut, označující identifikační číslo myši a 77 numerických atributů. Myši jsou rozděleny do osmi tříd podle jejich genotypu, chování a léčby.

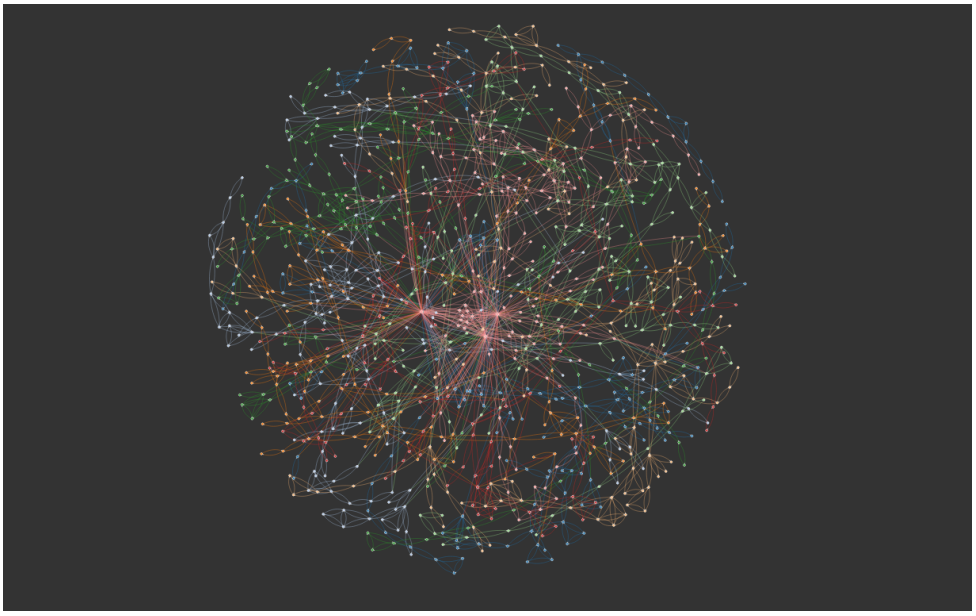
Vícevariační data jsou po jejich načtení převedena na síť pomocí algoritmů LRNet a ϵ-kNN, u kterého byl práh podobnosti nastaven na 0,98 a minimální počet sousedů na 1. Vzniklé grafy sítí jsou neorientované, ovšem neorientované hrany jsou reprezentovány dvěma orientovanými

hranami pro lepší viditelnost hran mezi komunitami. Vrcholy mají stejnou barvu a jsou umístěny blíže k sobě, pokud patří do stejné komunity. V Tabulce 9 je uvedena klasifikační přesnost zkonstruovaných grafů pomocí algoritmů LRNet a ϵ-kNN. Na Obrázku 20 je zobrazena výsledná síť LRNet Mice Protein Expression a na Obrázku 21 síť ϵ-kNN Mice Protein Expression. Na obou obrázcích jsou vrcholy rozděleny do komunit dle reálných tříd. Na Obrázku 22 je zobrazen sloupcový graf silhouette koeficientů pro jednotlivé vrcholy, rozdělené dle reálných tříd. Dle tohoto grafu lze usoudit, že vrcholy nejde jednoznačně přiřadit do tříd.

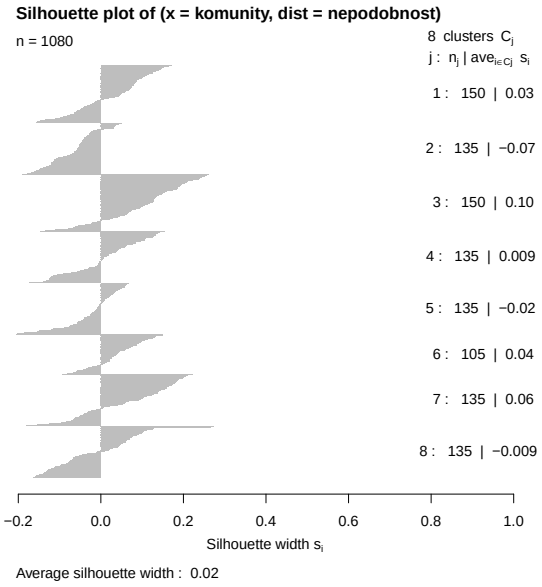
Graf	Vážená přesnost	Přesnost
Mice Protein Expression: LRNet	0,796	0,831
Mice Protein Expression: ϵ-kNN	0,938	0,934

Tabulka 9: Klasifikační přesnost zkonstruovaných grafů

Obrázek 20: LRNet Mice Protein Expression síť s reálnými třídami vrcholů



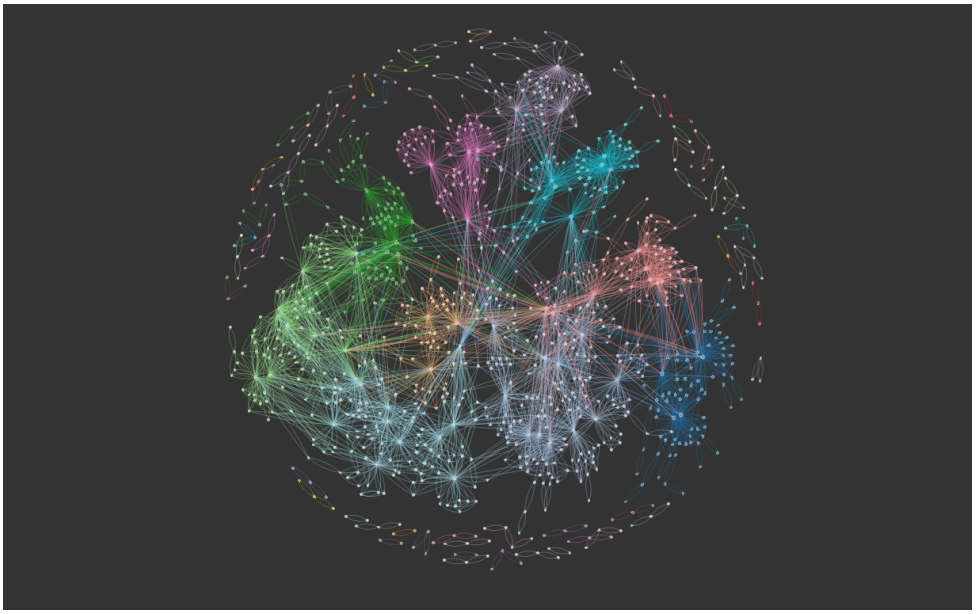
Obrázek 21: ϵ-kNN Mice Protein Expression síť s reálnými třídami vrcholů



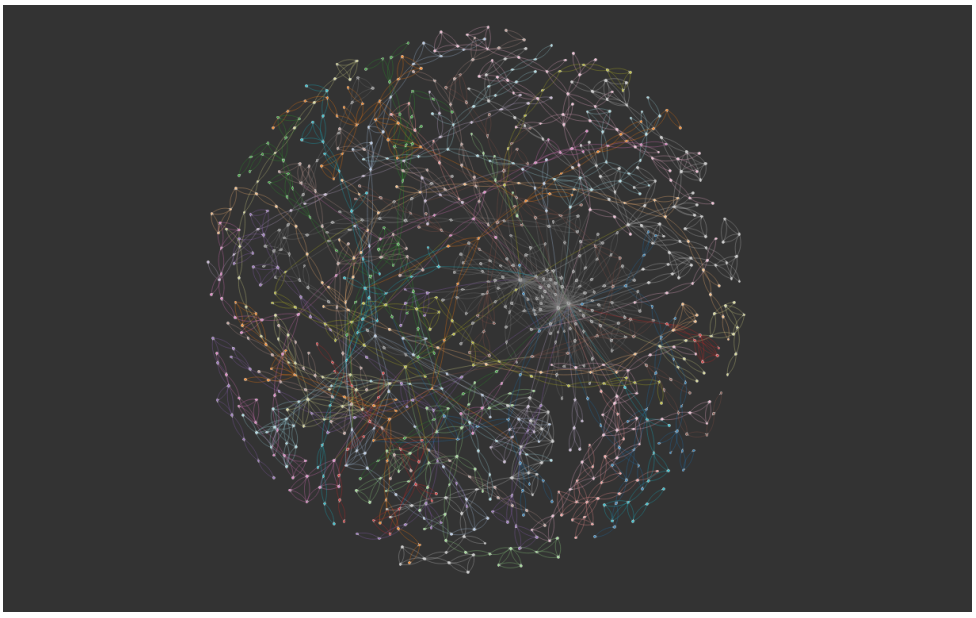
Obrázek 22: Silhouette koeficienty vrcholů rozdělených dle reálných tříd sítě Mice Protein Expression

Na Obrázcích 23 a 24 je zobrazena celá síť Mice Protein Expression po detekci komunit pomocí Louvain algoritmu. Na Obrázku 25 je zobrazen sloupcový graf silhouette koeficientů pro jednotlivé vrcholy sítě Mice Protein Expression po detekci komunit. V Tabulce 10 jsou uvedeny support a confidence hodnoty pro jednotlivé nalezené komunity na celé síti Mice Protein Ex-

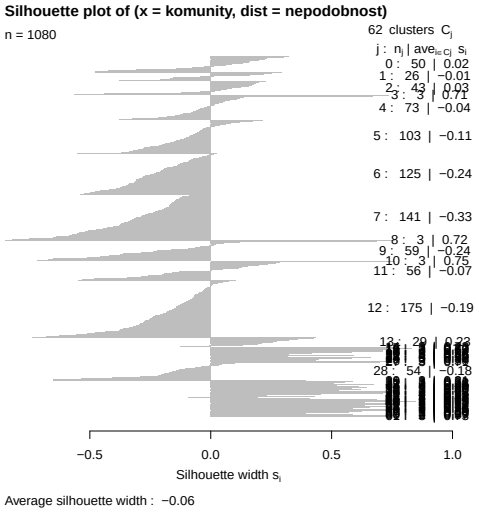
pression. Jelikož množství nalezených komunit je vysoké, tak jsou komunity s hodnotou support menší než 0,03 vynechány.



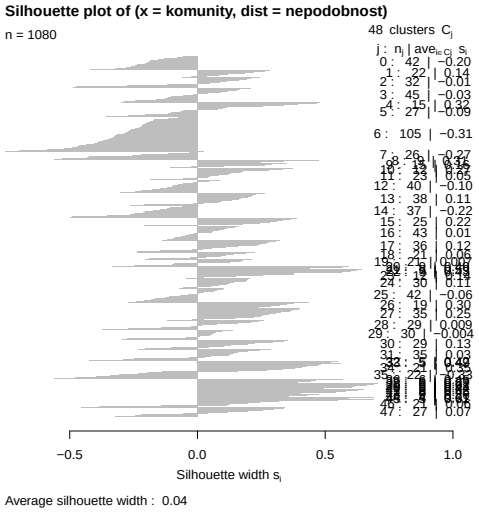
Obrázek 23: LRNet Mice Protein Expression síť s detekovanými komunitami



Obrázek 24: ϵ-kNN Mice Protein Expression síť s detekovanými komunitami



(a) LRNet Mice Protein Expression



(b) ϵ-kNN Mice Protein Expression

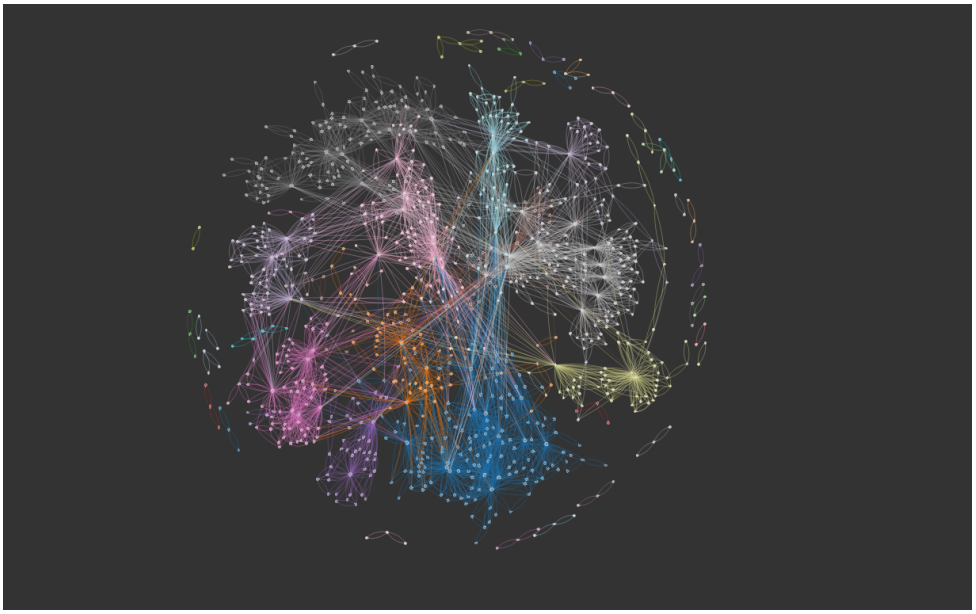
Obrázek 25: Silhouette koeficienty vrcholů detekovaných komunit na celé sítí Mice Protein Expression

Support LRNet Mice Protein Expression	Confidence LRNet Mice Protein Expression	Support ϵ-kNN Mice Protein Expression	Confidence ϵ-kNN Mice Protein Expression
0,162	0,382	0,097	0,343
0,130	0,454	0,041	1,000
0,115	0,480	0,039	0,535
0,095	0,592	0,038	0,643
0,067	0,630	0,038	0,643
0,054	0,475	0,037	0,875
0,051	0,482	0,035	0,500
0,050	0,648	0,034	0,486
0,046	0,620	0,033	0,667
0,039	0,558	0,032	1,000
-	-	0,032	1,000

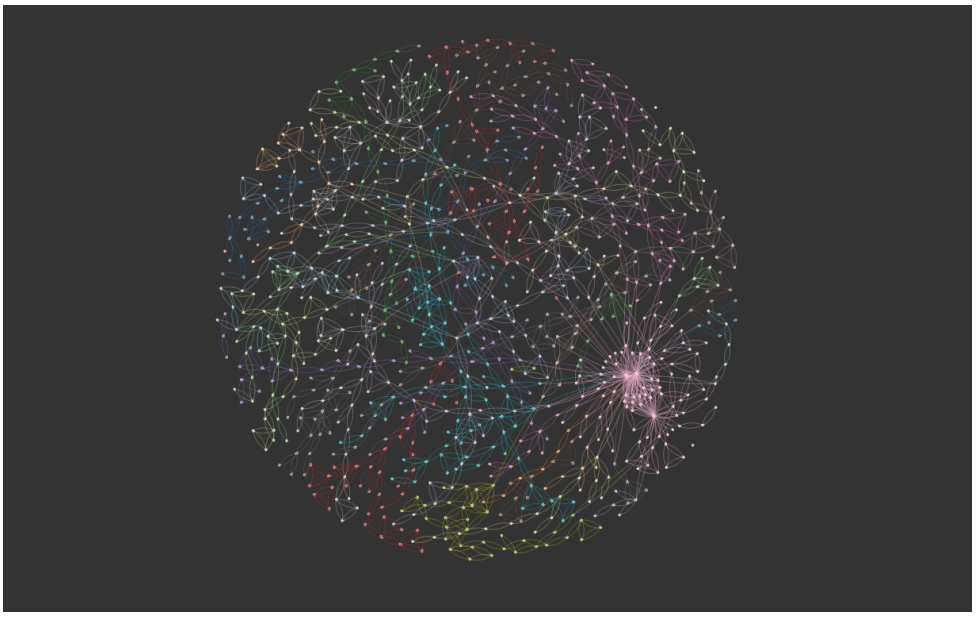
Tabulka 10: Support a confidence detekovaných komunit na celé síti Mice Protein Expression

Výsledné Mice Protein Expression sítě mají velké množství nesouvislých komponent, proto byla jejich část odfiltrována pomocí atributů *DYRK1AN* a *nNOSN*. Atribut *DYRK1AN* má rozsah od 0,145 do 2,516 a jeho horní hranice byla omezena na 1,520. Atribut *nNOSN* má rozsah od 0,010 do 0,260 a jeho horní hranice byla omezena na 0,220. Obě tato omezení způsobila odfiltrování 69 vrcholů. Filtrované sítě jsou zobrazeny na Obrázcích 26 a 27. Na Obrázku 28 je zobrazen sloupcový graf silhouette koeficientů jednotlivých vrcholů filtrované sítě a v Tabulce

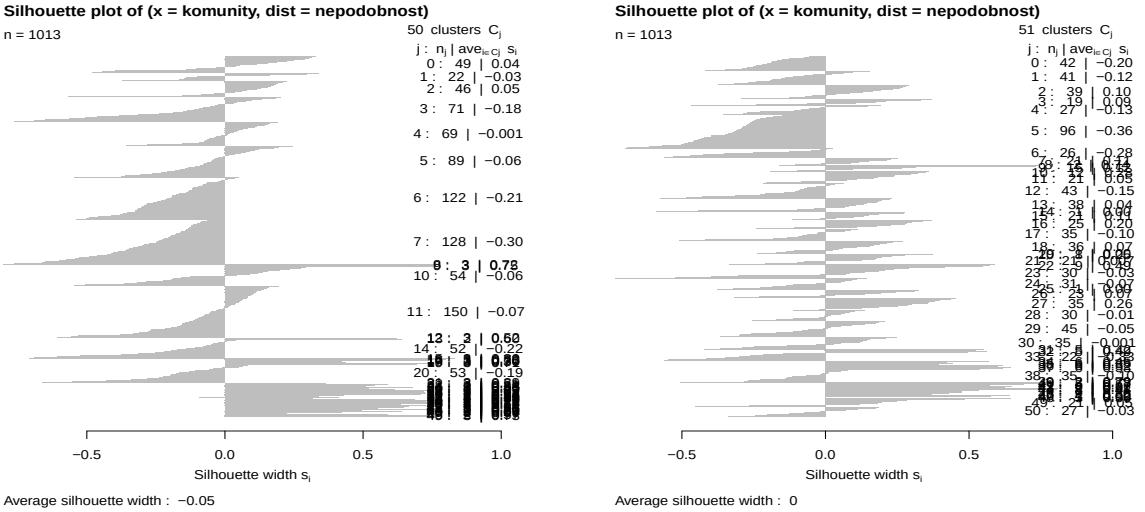
11 je uveden support a confidence pro jednotlivé komunity filtrované sítě. Jelikož množství nalezených komunit je vysoké, tak jsou komunity s hodnotou support menší než 0,03 vynechány.



Obrázek 26: Filtrovaná LRNet Mice Protein Expression síť s detekovanými komunitami



Obrázek 27: Filtrovaná ϵ-kNN Mice Protein Expression síť s detekovanými komunitami



(a) LRNet Mice Protein Expression

(b) ϵ-kNN Mice Protein Expression

Obrázek 28: Silhouette koeficienty vrcholů detekovaných komunit na filtrované síti Mice Protein Expression

Support LRNet Mice Protein Expression	Confidence LRNet Mice Protein Expression	Support ϵ-kNN Mice Protein Expression	Confidence ϵ-kNN Mice Protein Expression
0,148	0,533	0,094	0,323
0,124	0,476	0,044	0,667
0,121	0,463	0,042	0,744
0,087	0,636	0,041	0,643
0,065	0,621	0,040	0,366
0,056	0,474	0,038	1,000
0,056	0,667	0,038	0,500
0,053	0,500	0,036	0,667
0,051	0,442	0,035	0,571
0,047	0,604	0,035	1,000
0,045	0,587	0,035	1,000
-	-	0,035	0,600
-	-	0,031	0,516

Tabulka 11: Support a confidence detekovaných komunit na filtrované síti Mice Protein Expression

4.3.3 Audit Data

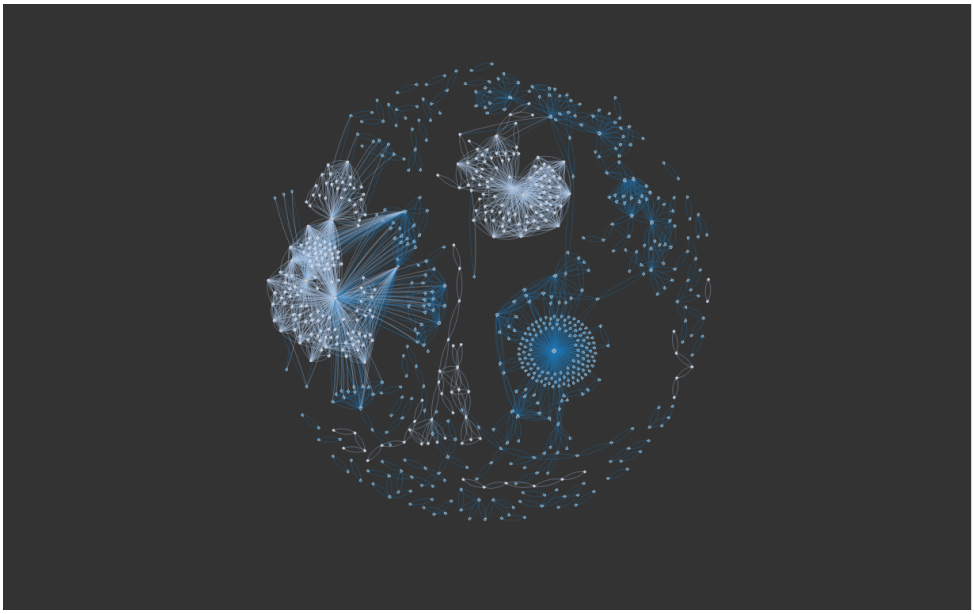
Datová sada Audit Data obsahuje data, která jsou určena pro klasifikaci podezřelých firem v Indii. Tato datová sada obsahuje 776 datových záznamů s vícevariačními daty. Každý záznam

má 17 numerických atributů. Firmy jsou rozděleny do dvou tříd podle toho, jestli jsou podezřelé nebo ne.

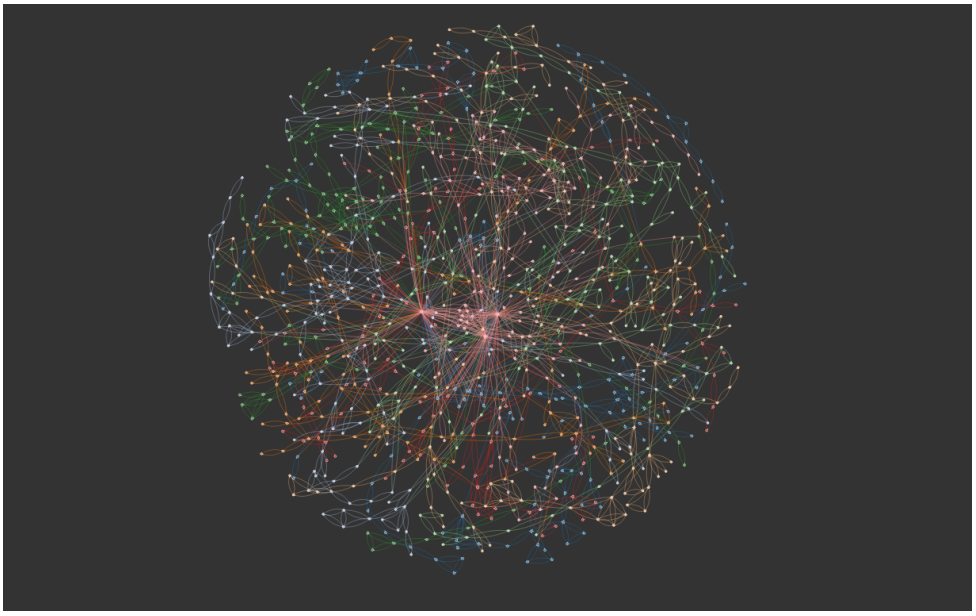
Vícevariační data jsou po jejich načtení převedena na síť pomocí algoritmů LRNet a ϵ-kNN, u kterého byl práh podobnosti nastaven na 0.98 a minimální počet sousedů na 1. Vzniklé grafy sítí jsou neorientované, ovšem neorientované hrany jsou reprezentovány dvěma orientovanými hranami pro lepší viditelnost hran mezi komunitami. Vrcholy mají stejnou barvu a jsou umístěny blíže k sobě, pokud patří do stejné komunity. V Tabulce 12 je uvedena klasifikační přesnost zkonstruovaných grafů pomocí algoritmů LRNet a ϵ-kNN. Na Obrázku 29 je zobrazena výsledná síť LRNet Audit Data a na Obrázku 30 síť ϵ-kNN Audit Data. Na obou obrázcích jsou vrcholy rozděleny do komunit dle reálných tříd. Na Obrázku 31 je zobrazen sloupcový graf silhouette koeficientů pro jednotlivé vrcholy, rozdělené dle reálných tříd. Dle tohoto grafu lze usoudit, že vrcholy nejde jednoznačně přiřadit do tříd.

Graf	Vážená přesnost	Přesnost
Audit Data: LRNet	0,959	0,983
Audit Data: ϵ-kNN	0,959	0,960

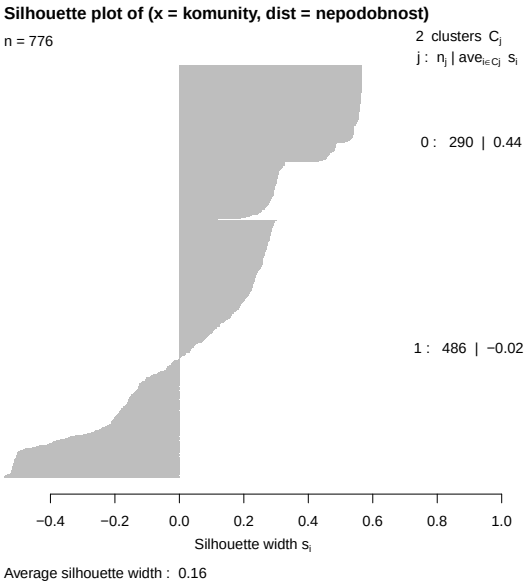
Tabulka 12: Klasifikační přesnost zkonstruovaných grafů



Obrázek 29: LRNet Audit Data síť s reálnými třídami vrcholů

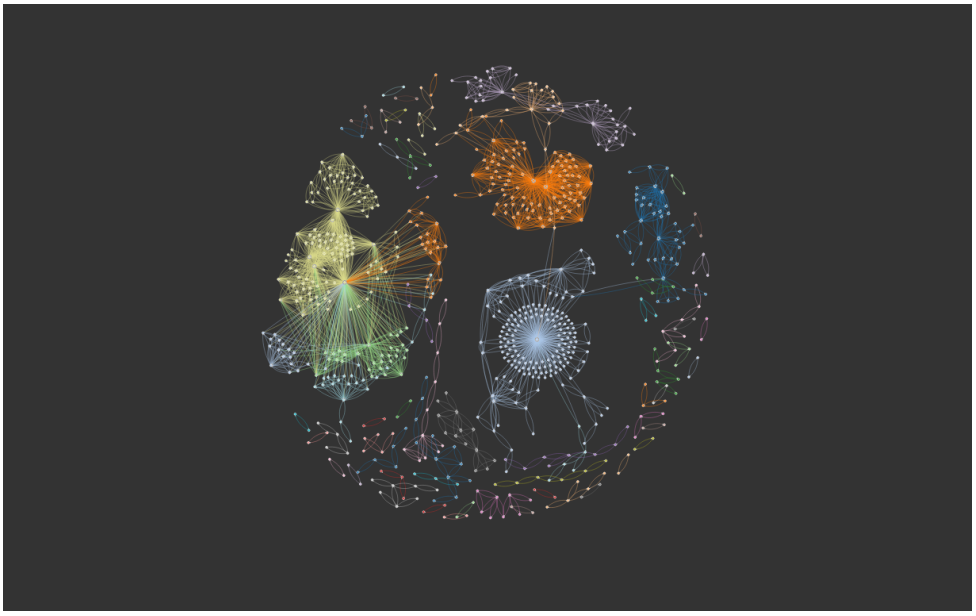


Obrázek 30: ϵ-kNN Audit Data síť s reálnými třídami vrcholů

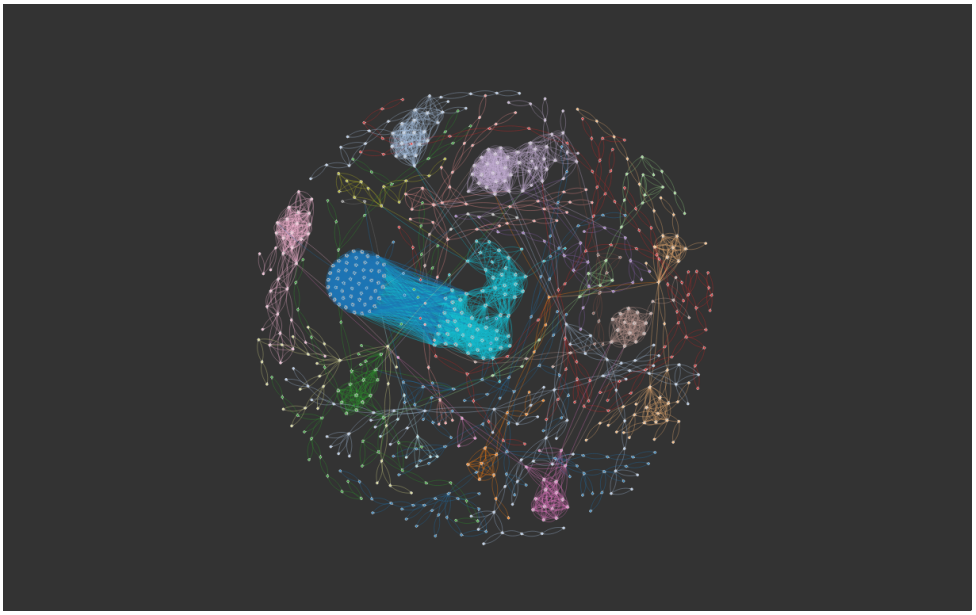


Obrázek 31: Silhouette koeficienty vrcholů rozdělených dle reálných tříd sítě Audit Data

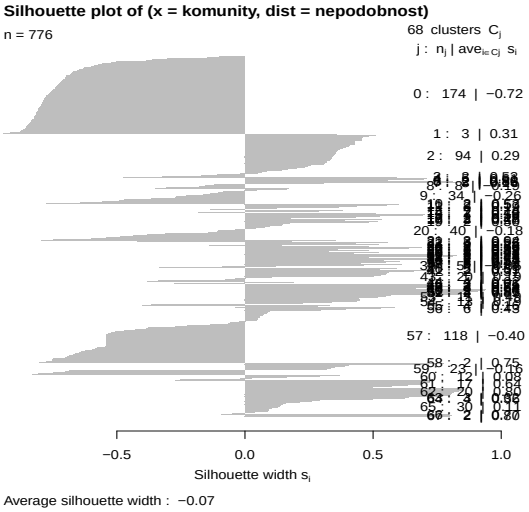
Na Obrázcích 32 a 33 je zobrazena celá síť Audit Data po detekci komunit pomocí Louvain algoritmu. Na Obrázku 34 je zobrazen sloupcový graf silhouette koeficientů pro jednotlivé vrcholy sítě Audit Data po detekci komunit. V Tabulce 13 jsou uvedeny support a confidence hodnoty pro jednotlivé nalezené komunity na celé síti Audit Data. Jelikož množství nalezených komunit je vysoké, tak jsou komunity s hodnotou support menší než 0,01 vynechány.



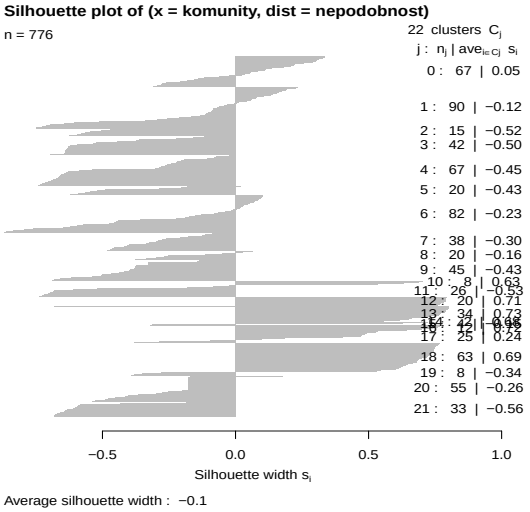
Obrázek 32: LRNet Audit Data síť s detekovanými komunitami



Obrázek 33: ϵ-kNN Audit Data síť s detekovanými komunitami



(a) LRNet Audit Data



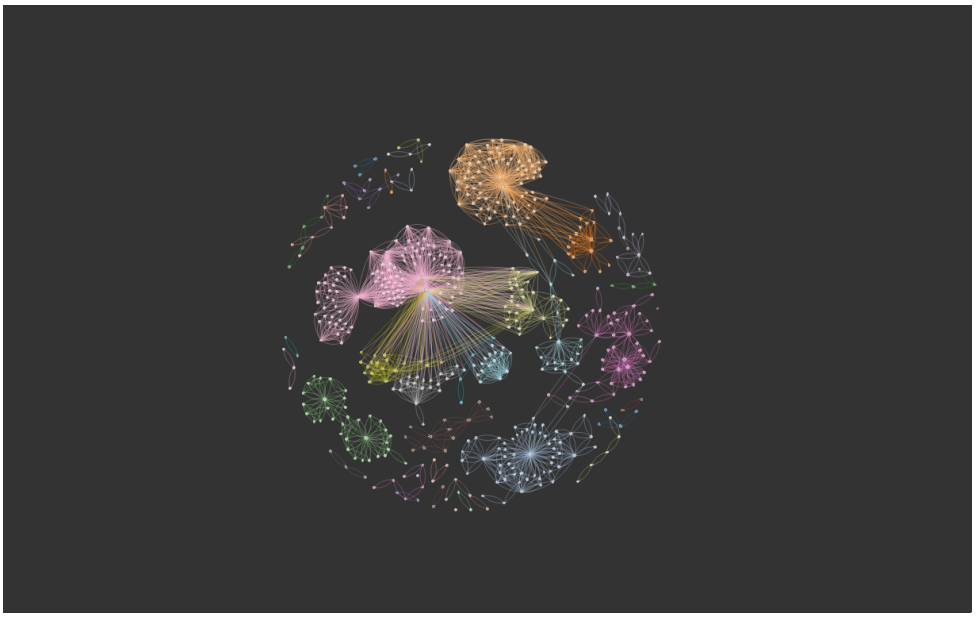
(b) ϵ-kNN Audit Data

Obrázek 34: Silhouette koeficienty vrcholů detekovaných komunit na celé síti Audit Data

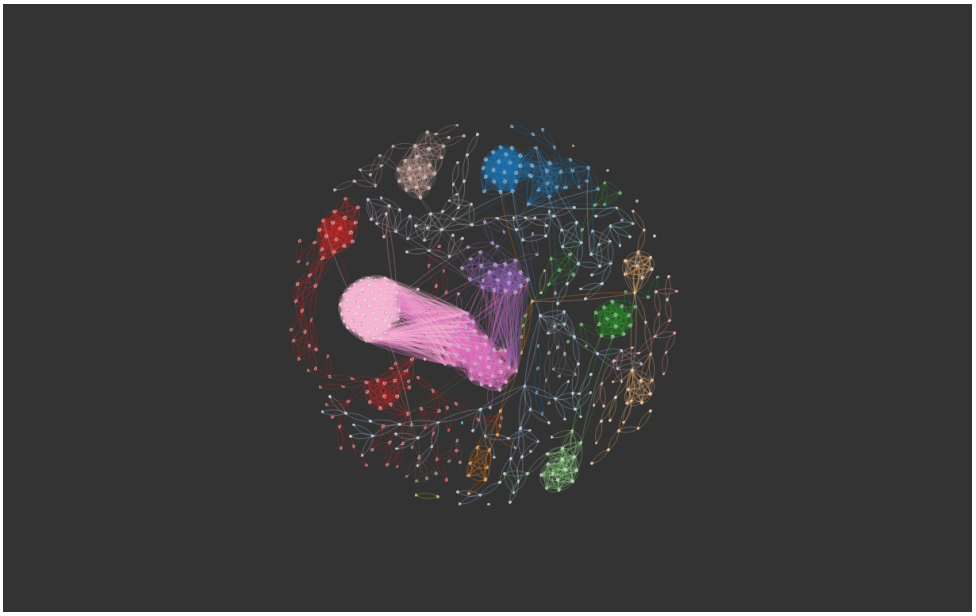
Support LRNet Audit Data	Confidence LRNet Audit Data	Support ϵ-kNN Audit Data	Confidence ϵ-kNN Audit Data
0,224	1,000	0,115	0,989
0,152	0,797	0,105	1,000
0,121	1,000	0,086	1,000
0,051	1,000	0,086	1,000
0,043	1,000	0,081	0,984
0,038	0,833	0,070	0,945
0,029	0,609	0,057	0,867
0,025	0,800	0,054	0,714
0,025	1,000	0,048	1,000
0,021	1,000	0,043	0,852
0,016	1,000	0,042	1,000
0,015	1,000	0,033	0,653
0,014	1,000	0,032	1,000
0,010	1,000	0,026	1,000
0,010	1,000	0,026	0,900
-	-	0,026	0,950
-	-	0,019	0,867
-	-	0,015	1,000
-	-	0,010	1,000
-	-	0,010	1,000

Tabulka 13: Support a confidence detekovaných komunit na celé síti Audit Data

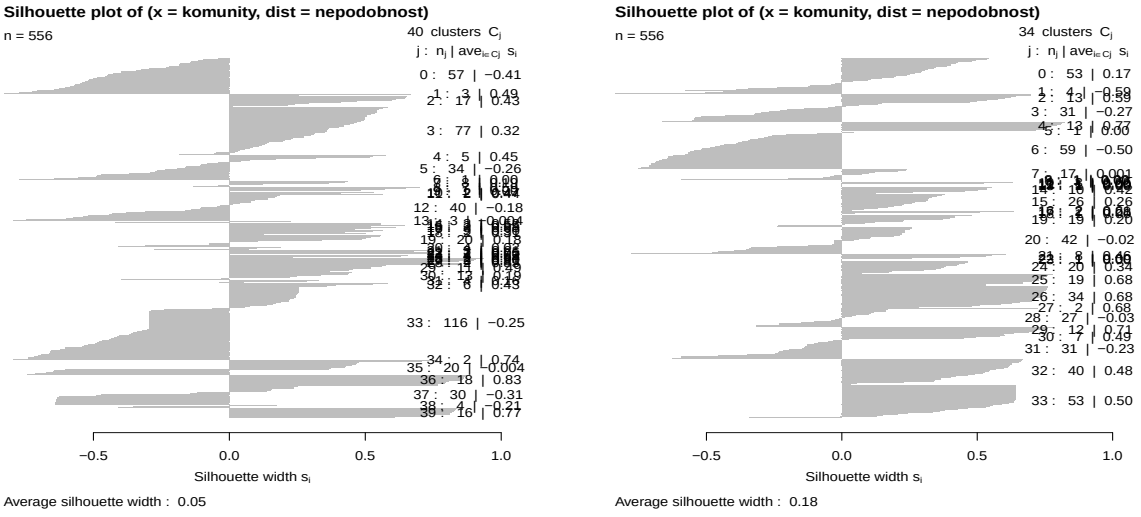
Výsledné Audit Data sítě mají velké množství nesouvislých komponent. Filtrace byla vyzkoušena pomocí atributu *SCORE*, který nejlépe rozděloval datovou sadu. Atribut *SCORE* má rozsah od 2,0 do 5,2 a jeho horní hranice byla omezena na 3,1. Toto omezení způsobilo odfiltrování 220 vrcholů. Filtrované sítě jsou zobrazeny na Obrázcích 35 a 36. Na Obrázku 37 je zobrazen sloupcový graf silhouette koeficientů jednotlivých vrcholů filtrované sítě a v Tabulce 14 je uveden support a confidence pro jednotlivé komunity filtrované sítě. Jelikož množství nalezených komunit je vysoké, tak jsou komunity s hodnotou support menší než 0,01 vynechány.



Obrázek 35: Filtrovaná LRNet Audit Data síť s detekovanými komunitami



Obrázek 36: Filtrovaná ϵ-kNN Audit Data síť s detekovanými komunitami



(a) LRNet Audit Data

(b) ϵ-kNN Audit Data

Obrázek 37: Silhouette koeficienty vrcholů detekovaných komunit na filtrované síti Audit Data

Support LRNet Audit Data	Confidence LRNet Audit Data	Support ϵ-kNN Audit Data	Confidence ϵ-kNN Audit Data
0,208	0,897	0,106	1,000
0,138	1,000	0,095	1,000
0,103	1,000	0,095	0,981
0,072	1,000	0,071	0,975
0,061	1,000	0,061	0,853
0,054	0,500	0,055	0,968
0,036	0,800	0,055	1,000
0,036	0,600	0,048	0,926
0,032	1,000	0,046	1,000
0,031	1,000	0,035	0,850
0,029	1,000	0,034	0,947
0,023	1,000	0,034	1,000
0,020	1,000	0,030	1,000
0,014	1,000	0,023	1,000
0,013	1,000	0,023	1,000
0,011	1,000	0,021	1,000
-	-	0,017	1,000
-	-	0,014	1,000
-	-	0,012	1,000

Tabulka 14: Support a confidence detekovaných komunit na filtrované síti Audit Data

Experimenty nám ukázaly, že algoritmy použité v tomto řešení ve většině případech dobře utvoří vztahy mezi záznamy stejných reálných tříd. Horší případ se vyskytl u datové sady Mice Protein Expression, kde hodnoty confidence detekovaných komunit na celé síti byly často velmi nízké a to především u algoritmu LRNet. Filtrace vrcholů může do určité míry přispět ke zlepšení kvality detekovaných komunit, ovšem přílišné kladení mezí na rozsah atributů může značně narušit strukturu sítě, což se ukázalo u experimentu s ϵ-kNN sítí datové sady Ecoli. Tyto experimenty prokázaly, že ověřený systém pro analýzu a vizualizaci vícevariačních sítí nám umožní snadnější a spolehlivější ověření jejich vlastností, které můžou vést k rychlejšímu řešení úloh z různých oblastí zájmu.

5 Závěr

Vícevariační sítě jsou všudypřítomné v široké škále oborů a proto zájem o jejich efektivní vizualizaci stále roste. Použití grafů je nejběžnějším přístupem vizualizace sítě, ale limitace na počet atributů, které mohou být snadno zakódovány na tuto strukturu, vede k popularizaci koordinovaných komponent a jiných metod, které jsou pro toto kódování více vhodné.

Cílem této práce bylo nalezení těchto metod či navržení nových, které umožní co nejpodrobnější, ale zároveň přehlednou explorační analýzu vícevariačních sítí. Tyto metody byly zakomponovány do webové aplikace, která dovoluje současně prozkoumávat a analyzovat síťovou topologii i vícevariační data skrze snadno srozumitelné uživatelské rozhraní. Toto umožňuje uživatelům snáze identifikovat odlehlá měření, vzory a trendy v kombinovaných datech. Tento průzkum je navíc podpořen funkcí detekce komunit, která automaticky rozdělí vrcholy do komunit. Uživatel může kromě této automatické detekce komunit využít manuálního rozřazování vrcholů do skupin. Rozložení vrcholů může být poté následně upraveno, podle toho zda vrcholy patří do stejné komunity (skupiny) či ne. Vrcholy stejných komunit (skupin) jsou umístěny blíže k sobě, což umožní jejich okamžitou identifikaci ve vizualizované síti. Pro vytvořené komunity (skupiny) se dále vytvoří síť s komunitami jako vrcholy, která je vizualizována pomocí grafu a umožňuje získat přehled o vztazích mezi těmito komunitami. Nástroj navíc nabízí uživateli možnost vrcholy odfiltrovávat dle jednotlivých atributů a tím prozkoumávat pouze část sítě a pozorovat změny v detekovaných komunitách na takto redukovaném grafu.

Systém by mohl být rozšířen a umožňovat kromě průzkumu atributů získaných z datových sad i výpočet derivovaných atributů, vycházejících z topologie sítě jako například vyhledání existence cesty mezi dvěma vrcholy, jejich vzdálenost či centrality. Kromě toho se nabízí nalézt řešení, jak rozložení grafu vypočítávat na straně serveru, jelikož tento výpočet probíhá v použitém systému na straně klientské a klade tak značné omezení na velikost sítě. Momentálně systém poskytuje náhled na atributy vrcholů, obdobně lze rozšířit systém tak, aby umožňoval průzkum atributů hran, které mohou hrát také důležitou roli k pochopení a hledání vzorů vícevariační sítě a s nimi i výpočet derivovaných atributů. Nakonec i doplnění funkcionality umožňující export vícevariační sítě a komunitní sítě do externích vizualizačních nebo editačních nástrojů jako Gephi a Adobe Illustrator, by umožnilo další úpravy a vylepšení.

Literatura

1. FRASINCAR, Flavius; TELEA, Alexandru; HOUBEN, Geert-Jan. Adapting graph visualization techniques for the visualization of RDF data. In: *Visualizing the semantic web*. Springer, 2006, s. 154–171.
2. SHEN, Zeqian; MA, Kwan-Liu; ELIASSI-RAD, Tina. Visual analysis of large heterogeneous social networks by semantic and structural abstraction. *IEEE transactions on visualization and computer graphics*. 2006, roč. 12, č. 6, s. 1427–1439.
3. TOMINSKI, Christian; ABELLO, James; SCHUMANN, Heidrun. CGV—An interactive graph visualization system. *Computers & Graphics*. 2009, roč. 33, č. 6, s. 660–678.
4. VAN DEN ELZEN, Stef; VAN WIJK, Jarke J. Multivariate network exploration and presentation: From detail to overview via selections and aggregations. *IEEE Transactions on Visualization and Computer Graphics*. 2014, roč. 20, č. 12, s. 2310–2319.
5. SHNEIDERMAN, Ben; ARIS, Aleks. Network visualization by semantic substrates. *IEEE transactions on visualization and computer graphics*. 2006, roč. 12, č. 5, s. 733–740.
6. JUSUFI, Ilir; DINGJIE, Yang; KERREN, Andreas. The network lens: Interactive exploration of multivariate networks using visual filtering. In: *2010 14th International Conference Information Visualisation*. 2010, s. 35–42.
7. TOMINSKI, Christian; ABELLO, James; VAN HAM, Frank; SCHUMANN, Heidrun. Fisheye tree views and lenses for graph visualization. In: *Tenth International Conference on Information Visualisation (IV'06)*. 2006, s. 17–24.
8. HURTER, Christophe; TISSOIRES, Benjamin; CONVERSY, Stéphane. Fromdady: Spreading aircraft trajectories across views to support iterative queries. *IEEE transactions on visualization and computer graphics*. 2009, roč. 15, č. 6, s. 1017–1024.
9. WU, Yingxin; TAKATSUKA, Masahiro. Visualizing multivariate network on the surface of a sphere. In: *Proceedings of the 2006 Asia-Pacific Symposium on Information Visualisation-Volume 60*. 2006, s. 77–83.
10. JUSUFI, Ilir; KERREN, Andreas; ZIMMER, Björn. Multivariate network exploration with JauntyNets. In: *2013 17th International Conference on Information Visualisation*. 2013, s. 19–27.
11. BEZERIANOS, Anastasia; CHEVALIER, Fanny; DRAGICEVIC, Pierre; ELMQVIST, Niklas; FEKETE, Jean-Daniel. Graphdice: A system for exploring multivariate social networks. In: *Computer Graphics Forum*. 2010, sv. 29, s. 863–872. Č. 3.
12. WATTENBERG, Martin. Visual exploration of multivariate graphs. In: *Proceedings of the SIGCHI conference on Human Factors in computing systems*. 2006, s. 811–819.

13. DUNNE, Cody; HENRY RICHE, Nathalie; LEE, Bongshin; METOYER, Ronald; ROBERTSON, George. GraphTrail: Analyzing large multivariate, heterogeneous networks while supporting exploration history. In: *Proceedings of the SIGCHI conference on human factors in computing systems*. 2012, s. 1663–1672.

14. NOBRE, Carolina; MEYER, Miriah; STREIT, Marc; LEX, Alexander. The state of the art in visualizing multivariate networks. In: *Computer Graphics Forum*. 2019, sv. 38, s. 807–832. Č. 3.

15. JUNKER, Björn H; KLUKAS, Christian; SCHREIBER, Falk. VANTED: a system for advanced data analysis and visualization in the context of biological networks. *BMC bioinformatics*. 2006, roč. 7, č. 1, s. 109.

16. AUBER, David; CHIRICOTA, Yves; JOURDAN, Fabien; MELANÇON, Guy. Multiscale visualization of small world networks. In: *IEEE Symposium on Information Visualization 2003 (IEEE Cat. No. 03TH8714)*. 2003, s. 75–81.

17. HEER, Jeffrey; BOYD, Danah. Vizster: Visualizing online social networks. In: *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.* 2005, s. 32–39.

18. NOBRE, Carolina. *OceanPaths: Visualizing Multivariate Oceanography Data*. 2016. Disertační práce.

19. PRETORIUS, A Johannes; VAN WIJK, Jarke J. Visual inspection of multivariate graphs. In: *Computer Graphics Forum*. 2008, sv. 27, s. 967–974. Č. 3.

20. SEDLMAIR, Michael; ISENBERG, Petra; BAUR, Dominikus; MAUERER, Michael; PIGORSCH, Christian; BUTZ, Andreas. Cardiogram: visual analytics for automotive engineers. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2011, s. 1727–1736.

21. SCOTT, John. Social network analysis. *Sociology*. 1988, roč. 22, č. 1, s. 109–127.

22. HENRY, Nathalie; FEKETE, Jean-Daniel. Matlink: Enhanced matrix visualization for analyzing social networks. In: *IFIP Conference on Human-Computer Interaction*. 2007, s. 288–302.

23. HENRY, Nathalie; FEKETE, Jean-Daniel. Matrixexplorer: a dual-representation system to explore social networks. *IEEE transactions on visualization and computer graphics*. 2006, roč. 12, č. 5, s. 677–684.

24. HENRY, Nathalie; FEKETE, Jean-Daniel; MCGUFFIN, Michael J. Nodetrix: a hybrid visualization of social networks. *IEEE transactions on visualization and computer graphics*. 2007, roč. 13, č. 6, s. 1302–1309.

25. GHANI, Sohaib; KWON, Bum Chul; LEE, Seungyoon; YI, Ji Soo; ELMQVIST, Niklas. Visual analytics for multimodal social network analysis: A design study with social scientists. *IEEE transactions on visualization and computer graphics*. 2013, roč. 19, č. 12, s. 2032–2041.

26. SCHREIBER, Falk; DWYER, Tim; MARRIOTT, Kim; WYBROW, Michael. A generic algorithm for layout of biological networks. *BMC bioinformatics*. 2009, roč. 10, č. 1, s. 375.

27. KARP, Peter D; PALEY, Suzanne. Automated drawing of metabolic pathways. In: *Proceedings of the 3rd International Conference on Bioinformatics and Genome Research*. 1995, s. 225–238.

28. PARTL, Christian; LEX, Alexander; STREIT, Marc; KALKOFEN, Denis; KASHOFER, Karl; SCHMALSTIEG, Dieter. enRoute: Dynamic path extraction from biological pathway maps for in-depth experimental data analysis. In: *2012 IEEE Symposium on Biological Data Visualization (BioVis)*. 2012, s. 107–114.

29. LEX, Alexander; PARTL, Christian; KALKOFEN, Denis; STREIT, Marc; GRATZL, Samuel; WASSERMANN, Anne Mai; SCHMALSTIEG, Dieter; PFISTER, Hanspeter. Entourage: Visualizing relationships between biological pathways using contextual subsets. *IEEE transactions on visualization and computer graphics*. 2013, roč. 19, č. 12, s. 2536–2545.

30. MEYER, Miriah; WONG, Bang; STYCZYNSKI, Mark; MUNZNER, Tamara; PFISTER, Hanspeter. Pathline: A tool for comparative functional genomics. In: *Computer Graphics Forum*. 2010, sv. 29, s. 1043–1052. Č. 3.

31. BARSKY, Aaron; MUNZNER, Tamara; GARDY, Jennifer; KINCAID, Robert. Cerebral: Visualizing multiple experimental conditions on a graph with biological context. *IEEE transactions on visualization and computer graphics*. 2008, roč. 14, č. 6, s. 1253–1260.

32. NEW, Joshua; KENDALL, Wesley; HUANG, Jian; CHESLER, Elissa. Dynamic visualization of coexpression in systems genetics data. *IEEE transactions on visualization and computer graphics*. 2008, roč. 14, č. 5, s. 1081–1095.

33. DIEHL, S; TELEA, AC. Multivariate Graphs in Software Engineering.

34. BYELAS, Heorhiy; TELEA, Alexandru. Visualizing multivariate attributes on software diagrams. In: *2009 13th European Conference on Software Maintenance and Reengineering*. 2009, s. 335–338.

35. BALL, Robert; FINK, Glenn A; NORTH, Chris. Home-centric visualization of network traffic for security administration. In: *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*. 2004, s. 55–64.

36. PEARLMAN, Jason; RHEINGANS, Penny. Visualizing network security events using compound glyphs from a service-oriented perspective. In: *VizSEC 2007*. Springer, 2008, s. 131–146.

37. YIN, Xiaoxin; YURCIK, William; TREASTER, Michael; LI, Yifan; LAKKARAJU, Kiran. VisFlowConnect: netflow visualizations of link relationships for security situational awareness. In: *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*. 2004, s. 26–34.

38. PEYSAKHOVICH, Vsevolod; HURTER, Christophe; TELEA, Alexandru. Attribute-driven edge bundling for general graphs with applications in trail analysis. In: *2015 IEEE Pacific Visualization Symposium (PacificVis)*. 2015, s. 39–46.

39. WOLFF, Alexander. Drawing subway maps: A survey. *Informatik-Forschung und Entwicklung*. 2007, roč. 22, č. 1, s. 23–44.

40. AMAR, Robert; EAGAN, James; STASKO, John. Low-level components of analytic activity in information visualization. In: *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.* 2005, s. 111–117.

41. VALIATI, Eliane RA; PIMENTA, Marcelo S; FREITAS, Carla MDS. A taxonomy of tasks for guiding the evaluation of multidimensional visualizations. In: *Proceedings of the 2006 AVI workshop on Beyond time and errors: novel evaluation methods for information visualization*. 2006, s. 1–6.

42. LEE, Bongshin; PLAISANT, Catherine; PARR, Cynthia Sims; FEKETE, Jean-Daniel; HENRY, Nathalie. Task taxonomy for graph visualization. In: *Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization*. 2006, s. 1–5.

43. PRETORIUS, Johannes; PURCHASE, Helen C; STASKO, John T. Tasks for multivariate network analysis. In: *Multivariate Network Visualization*. Springer, 2014, s. 77–95.

44. KERREN, Andreas; PURCHASE, Helen C; WARD, Matthew O. Introduction to multivariate network visualization. In: *Multivariate Network Visualization*. Springer, 2014, s. 1–9.

45. JUSUFI, Ilir. *Multivariate networks: visualization and interaction techniques*. 2013. Disertační práce.

46. KOLOSOVSKIY, Maxim A. Data structure for representing a graph: combination of linked list and hash table. *arXiv preprint arXiv:0908.3089*. 2009.

47. JASSIM, Wadhah S. Incidence Matrices of Directed Graphs of Groups and their up-down Pregroups. *Sultan Qaboos University Journal for Science [SQUJS]*. 2017, roč. 22, č. 1, s. 40–47.

48. SOMMER, Christian. *Graph Representation in Memory* [online]. 2010 [cit. 2020-02-13]. Dostupné z: `http://www.sommer.jp/aa10/aa8.pdf`.

49. OCHODKOVA, Eliska; ZEHNALOVA, Sarka; KUDELKA, Milos. Graph construction based on local representativeness. In: *International Computing and Combinatorics Conference*. 2017, s. 654–665.

50. BENTLEY, Jon L; STANAT, Donald F; WILLIAMS JR, E Hollins. The complexity of finding fixed-radius near neighbors. *Information processing letters*. 1977, roč. 6, č. 6, s. 209–212.

51. CHAZELLE, Bernard. An improved algorithm for the fixed-radius neighbor problem. *Information Processing Letters*. 1983, roč. 16, č. 4, s. 193–198.

52. CHEN, Jie; FANG, Haw-ren; SAAD, Yousef. Fast approximate kNN graph construction for high dimensional data via recursive Lanczos bisection. *Journal of Machine Learning Research*. 2009, roč. 10, č. Sep, s. 1989–2012.

53. DONG, Wei; MOSES, Charikar; LI, Kai. Efficient k-nearest neighbor graph construction for generic similarity measures. In: *Proceedings of the 20th international conference on World wide web*. 2011, s. 577–586.

54. SCHÖFFEL, Sebastian; SCHWANK, Johannes; EBERT, Achim. A user study on multivariate edge visualizations for graph-based visual analysis tasks. In: *2016 20th International Conference Information Visualisation (IV)*. 2016, s. 165–170.

55. KO, Sungahnn; AFZAL, Shehzad; WALTON, Simon; YANG, Yang; CHAE, Junghoon; MALIK, Abish; JANG, Yun; CHEN, Min; EBERT, David. Analyzing high-dimensional multivariate network links with integrated anomaly detection, highlighting and exploration. In: *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*. 2014, s. 83–92.

56. NIELSEN, Cydney B; JACKMAN, Shaun D; BIROL, Inanç; JONES, Steven JM. ABySS-Explorer: visualizing genome sequence assemblies. *IEEE transactions on visualization and computer graphics*. 2009, roč. 15, č. 6, s. 881–888.

57. SHANNON, Ross; HOLLAND, Thomas; QUIGLEY, Aaron. Multivariate graph drawing using parallel coordinate visualisations. *University College Dublin, School of Computer Science and Informatics, Tech. Rep*. 2008, roč. 6, s. 2008.

58. JAVED, Waqas; ELMQVIST, Niklas. Exploring the design space of composite visualization. In: *2012 ieee pacific visualization symposium*. 2012, s. 1–8.

59. PLAISANT, Catherine; SHNEIDERMAN, Ben; MUSHLIN, Rich. An information architecture to support the visualization of personal histories. *Information Processing & Management*. 1998, roč. 34, č. 5, s. 581–597.

60. MATKOVIC, Kresimir; FREILER, Wolfgang; GRACANIN, Denis; HAUSER, Helwig. Comvis: A coordinated multiple views system for prototyping new visualization technology. In: *2008 12th international conference information visualisation*. 2008, s. 215–220.

61. LOKUGE, Ishantha; ISHIZAKI, Suguru. Geospace: An interactive visualization system for exploring complex information spaces. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. 1995, s. 409–414.

62. FORTUNATO, Santo. Community detection in graphs. *Physics reports*. 2010, roč. 486, č. 3-5, s. 75–174.

63. BARABÁSI, Albert-László et al. *Network science*. Cambridge university press, 2016.

64. BLONDEL, Vincent D; GUILLAUME, Jean-Loup; LAMBIOTTE, Renaud; LEFEBVRE, Etienne. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*. 2008, roč. 2008, č. 10, s. P10008.

65. BOSTOCK, Mike. *Data-Driven Documents* [online]. 2019 [cit. 2020-05-03]. Dostupné z: `https://d3js.org/`.

66. *Data-Driven Documents* [online]. 2020 [cit. 2020-05-03]. Dostupné z: `https://www.newtonsoft.com/json`.

67. DUA, Dheeru; GRAFF, Casey. *UCI Machine Learning Repository*. 2017. Dostupné také z: `http://archive.ics.uci.edu/ml`.

68. ROUSSEEUW, Peter J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*. 1987, roč. 20, s. 53–65.